



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE INGENIERÍA

**SISTEMA DE LOCALIZACIÓN DE OBJETOS
EN ESPACIOS CERRADOS
POR MEDIO DE RF**

TESIS

**QUE PARA OBTENER EL GRADO DE
INGENIERO EN TELECOMUNICACIONES**

PRESENTA:

MIGUEL ÁNGEL ALONSO RIOS

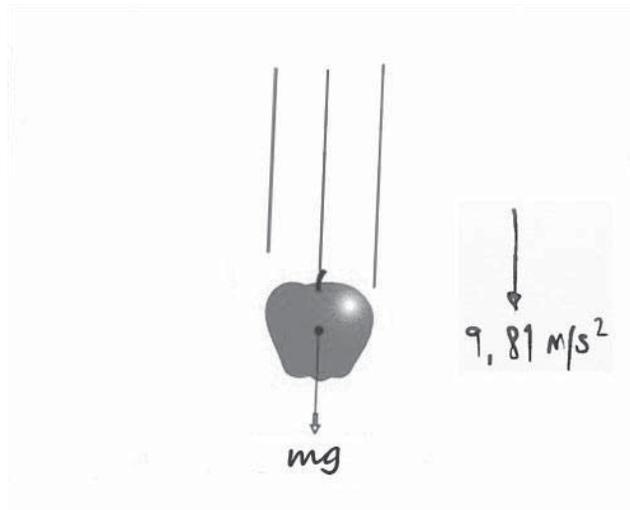
DIRECTOR DE TESIS:

M.I. JUVENTINO CUÉLLAR GONZÁLEZ



CIUDAD UNIVERSITARIA, MÉXICO D. F., 2012.

*A mis padres, hermana y novia, que sufrieron mis ausencias,
a mis amigos cercanos que nunca dejaron de serlo,
a todos ellos que han estado siempre a mi lado,
gracias por hacer esto más fácil.*



*"Si he conseguido ver más lejos, es porque me he
parado sobre los hombros de gigantes"*

Isaac Newton

Agradecimientos

La culminación de esta tesis constituye un gran paso hacia adelante en mi formación profesional como Ingeniero.

Quiero dedicar este trabajo a dos personas sin las cuales esto nunca hubiera sido posible, a mis Padres: Juanita y Miguel Ángel. Gracias a su apoyo incondicional y confianza he podido llegar a este punto.

Agradezco la ayuda prestada a todas las personas que han colaborado durante mi desarrollo personal y profesional, así como a aquellas que me han acompañado durante esta etapa de mi vida, su apoyo ha hecho de mi lo que soy.

Quiero dar las gracias a tres personas cuya contribución ha sido fundamental para el enriquecimiento de este trabajo: a la Ing. María del Rosario Barragán Paz, a la Ing. Margarita Bautista González y al Dr. Ignacio Flores Llamas.

Asimismo, quiero agradecer también la dedicación y atención mostrada durante todo el proyecto a mi tutor, M.I. Juventino Cuéllar González, no tan solo por su ayuda, sino por confiar en mí a la hora de ofrecerme la posibilidad de realizar este proyecto.

Y por último, y no por ello en menor medida, quiero agradecer a la **Universidad Nacional Autónoma de México UNAM** por haberme brindado una educación integral y de Calidad.

Miguel Ángel Alonso Rios.

Tabla de contenido

CAPÍTULO	PÁGINA
Resumen.....	3
Abstract.....	4
Agradecimientos.....	7
Tabla de contenido.....	9
Índice de figuras y tablas.....	13
Glosario.....	16
1. Introducción.....	19
1.1 Objetivo.....	19
1.2 Motivación.....	20
1.3 Antecedentes.....	21
1.4 Organización del trabajo.....	24
2. Sistemas de localización de objetos.....	25
2.1 Sistemas de localización en espacios abiertos.....	26
2.1.1 Sistema de Posicionamiento Global.....	26
2.1.2 RADAR (Radio Detection And Ranging).....	27
2.1.3 Localización usando telefonía móvil.....	28
2.1.4 GPS Asistido (Assisted Global Positioning System).....	28
2.2 Sistemas de localización en espacios cerrados.....	29
2.2.1 Identificación por Radiofrecuencia RFID.....	30
2.2.2 Localización por Infrarrojos.....	31
2.2.3 Bluetooth.....	31
2.2.4 Wi-Fi (estándar IEEE 802.11).....	32

2.2.5	UWB (Ultra Wideband)	33
2.2.6	ZigBee	33
2.2.7	Elección de tecnología inalámbrica	34
2.3	Técnicas de estimación de distancias	35
2.3.1	Tiempo de llegada (Time of Arrival ToA)	35
2.3.2	Diferencia de tiempos de llegada (Time Difference of Arrival TDoA)	36
2.3.3	Ángulo de llegada (Angle of Arrival AoA).....	37
2.3.4	Tiempo de Vuelo (Time of Flight ToF)	39
2.3.5	Indicador de Intensidad de Señal Recibida	40
2.3.6	Elección de la técnica de localización.....	42
3.	Indicador de Potencia Recibida RSSI	45
3.1	Características y especificaciones	46
3.2	Métodos de estimación de localización basados en RSSI	49
3.2.1	Triangulación	49
3.2.2	Reconocimiento de patrones	51
4.	Descripción del protocolo estándar IEEE 802.15.4	53
4.1	Características y especificaciones	53
4.1.1	Capa Física	55
4.1.2	Capa de enlace.....	57
4.2	ZigBee.....	59
4.2.1	Nivel de red	62
4.2.2	Nivel de aplicación.....	62
4.2.3	Dispositivos ZigBee	63
4.2.4	Seguridad	65
5.	Diseño del hardware	67
5.1	Módulos XBee.....	67

5.1.1	Modos de operación	72
5.1.2	Comunicación Serial.....	72
5.1.3	Operación transparente	74
5.1.4	Operación API	74
5.1.5	Petición de envío (TX Request) para direcciones de 16 bits	76
5.1.6	Recepción de datos (RX Packet) para direcciones de 16 bits	78
5.1.7	Estructura de una trama de estado de paquete.....	79
5.1.8	Direccionamiento ZigBee en XBee.....	80
5.2	Determinación de módulos a implementar.....	81
5.3	Interfaz para XBee	81
5.4	Desarrollo programa para el Microcontrolador	82
5.5	Microcontroladores.....	82
5.5.1	Microcontrolador elegido	86
5.6	Alcances y limitaciones	87
5.7	Desarrollo del proyecto	88
5.7.1	Funcionamiento del sistema	89
5.7.2	Infraestructura.....	91
5.7.3	Módulo remoto.....	93
5.7.4	Nodos Balizas.....	97
5.7.5	Hardware en la computadora receptora	101
5.7.6	Software de configuración	103
5.7.7	Arquitectura de la red	104
6.	Diseño del software.....	109
6.1	Lenguaje de programación.....	110
6.2	Java.....	111
6.3	Algoritmo	113

6.4	Interfaz	117
6.5	Gestión de tramas	118
6.6	Escáner para la red ZigBee.....	120
6.7	Descripción de la aplicación.....	123
6.7.1	Nuevo proyecto.....	123
6.7.2	Registro de dispositivos que conforman la red	124
6.7.3	Levantamiento (<i>fingerprinting</i>).....	125
6.7.4	Localización	127
7.	Pruebas y resultados	129
7.1	Colocación de los nodos baliza.....	131
7.2	Mapa de cobertura	133
7.3	Consideraciones / limitaciones.....	134
8.	Conclusiones.....	135
8.1	Líneas futuras de trabajo.....	139
9.	Bibliografía	141
Anexos	142

Índice de figuras y tablas

Lista de Figuras

Figura 2.1. Esquema de la localización por GPS	26
Figura 2.2. Sistema A-GPS	29
Figura 2.3. Tiempo de Llegada.....	35
Figura 2.4. Estimación de la distancia por la diferencia de tiempos de llegada. ...	37
Figura 2.5. Método de Ángulo de Llegada.....	38
Figura 2.6. Estimación de la distancia por el ángulo de llegada.....	38
Figura 2.7. Forma en la que se determina la posición del tag con ToF.....	39
Figura 2.8. Estimación de la ubicación con RSSI.....	42
Figura 3.1. Reflexión.	48
Figura 3.2. Difracción.	48
Figura 3.3. Dispersión de una onda.	48
Figura 3.4. Estimación de la localización utilizando triangulación.	50
Figura 3.5. Patrón de radiación de antena omnidireccional	50
Figura 4.1. Protocolo IEEE 802.15.4	54
Figura 4.2. Canales de Comunicación IEEE 802.15.4.	56
Figura 4.3. Pila de protocolos IEEE 802.15.4.....	58
Figura 4.4. Aplicaciones ZigBee.....	59
Figura 4.5. Comparación de tecnologías Inalámbricas.	60
Figura 4.6. ZigBee Stack.....	61
Figura 4.7. Dispositivos FFD y RFD.	64
Figura 5.1. Módulo XBee.....	68
Figura 5.2. XBee/XBee-Pro.....	68
Figura 5.3. Tipos de antenas.....	71
Figura 5.4. Comunicación UART.....	73
Figura 5.5. Paquete de datos UART, transmitido a través del módulo de RF.....	73
Figura 5.6. Estructura general de una trama en modo de operación API.....	75
Figura 5.7. Estructura para el envío en modo API usando 16 bits.	77

Figura 5.8. Estructura para la recepción en modo API usando 16 bits.....	78
Figura 5.9. Trama que indica el estado del paquete enviado	79
Figura 5.10. Desarrollo de software.	85
Figura 5.11. Proceso de programación del MCU.	86
Figura 5.12. Microcontrolador.....	87
Figura 5.13. Diagrama a bloques.	89
Figura 5.14. Sistema de localización.....	90
Figura 5.15. Infraestructura de la red ZigBee.....	92
Figura 5.16. Diagrama de flujo del dispositivo remoto.....	94
Figura 5.17. Diagrama de flujo del dispositivo remoto.....	94
Figura 5.18. Conexión UART entre el módulo XBee y el microcontrolador.....	95
Figura 5.19. Configuración del oscilador de cristal.....	96
Figura 5.20. Esquemático del dispositivo remoto.	96
Figura 5.21. Conexión para la transmisión.....	98
Figura 5.22. Conexión para la recepción.....	99
Figura 5.23. Diagrama de flujo, Nodo Baliza.....	100
Figura 5.24. Implementación del dispositivo receptor.	102
Figura 5.25. Software X-CTU.	104
Figura 5.26. Esquema general de la aplicación.....	107
Figura 6.1. Compilación y ejecución de un programa java.....	113
Figura 6.2. Recepción del vector de potencia.	114
Figura 6.3 Inicio de la aplicación.	123
Figura 6.4 Captura de parámetros de configuración de los dispositivos.	124
Figura 6.5 Realización del proceso de Fingerprinting.	126
Figura 6.6 Localización del dispositivo remoto.....	127
Figura 7.1 Plano de la Planta.....	129
Figura 7.2 Posiciones del dispositivo remoto.	130
Figura 7.3 Puntos de escaneo.....	133
Figura 7.4 Limitaciones de la aplicación.....	134

Lista de Tablas

Tabla 4-1 Bandas de Frecuencia para el estándar IEEE 802.15.4.....	56
Tabla 4-2. Frecuencia de los canales IEEE 802.15.4.....	56
Tabla 5-1. Especificaciones para XBee/XBee-PRO OEM RF Modules.....	69
Tabla 5-2. Descripciones de los pines para XBee and XBee-PRO Modules.....	70
Tabla 5-3. Identificador API y su tipo de mensaje.	76
Tabla 5-4. Dispositivos de la red.	105
Tabla 5-5. Configuración de los módulos.	106
Tabla 6-1. Tabla generada en el levantamiento.	115
Tabla 7-1 Vectores de Potencia.	133

Glosario

802.11	Conjunto de protocolos de comunicaciones inalámbricas
ACK	Reconocimiento (<i>Acknowledgement</i>)
AES	Advanced Encryption Standard
AP	Punto de acceso (<i>Access Point</i>)
API	Interfaz de programación de aplicaciones (<i>Application Programming Interface</i>)
Beacon	Dato que transmite un punto de acceso para notificar su presencia
CPU	Unidad central de procesamiento (<i>Central Processing Unit</i>)
CSMA/CA	Acceso al medio inalámbrico con evasión de colisiones (<i>Carrier Sense Multiple Access with Collision Avoidance</i>)
Driver	Software que hace de interfaz entre un dispositivo físico y el Sistema Operativo
DSSS	<i>Direct Sequence Spread Spectrum</i>
ED	Dispositivo final (<i>End Device</i>)
EEPROM	Memoria de solo lectura programable y borrable eléctricamente (<i>Electrically-Erasable Programmable Read-Only Memory</i>)
FFD	Dispositivo de funcionalidad completa (<i>Full-Function Device</i>)
GUI	Interfaz gráfica de usuario (<i>Graphical User Interface</i>)
IDE	<i>Integrated Development Environment</i>
IEEE	Institute of Electrical and Electronics Engineers
ISM	Banda libre de frecuencia usada para fines industriales, científicos y médicos
JVM	Máquina Virtual Java (<i>Java Virtual Machine</i>)
LAN	Red de Área Local (Local Area Network)
LLC	Control de enlace lógico (<i>Logical Link Control</i>)
LOS	Line of sight, Enlace de radio que debe tener visibilidad directa entre antenas
LR-WPAN	Redes Inalámbricas de Área Personal de Baja Velocidad (<i>Low-Rate Wireless Personal Area Networks</i>)
LSB	Bit menos significativo (<i>Less Significant Bit</i>)
MAC	Capa de control de acceso al medio (<i>Medium Access Control layer</i>)
MSB	Bit más significativo (<i>Most Significant Bit</i>)
OQPSK	Offset Quadrature Phase Shift Keying
OSI	Open Systems Interconnection
PAN	Red de área personal (<i>Personal Area Network</i>)
PCB	Tarjeta del circuito impreso (<i>Printed Circuit Board</i>)
PDA	<i>Personal Digital Assistant</i>
PHY	Capa física (<i>Physical layer</i>)
RF	Radio frecuencia (<i>Radio Frequency</i>)

RFD	Dispositivo de funcionalidad reducida (<i>Reduced-Function Device</i>)
RFID	Identificación por radiofrecuencia (<i>Radio Frequency Identification</i>)
RSSI	Indicador de potencia de señal recibida (<i>Received Signal Strength Indicator</i>)
SPI	Serial Peripheral Interface
SRLI	Sistemas de Radiolocalización en interiores
Thread	Hilo de ejecución de un programa
TM	Terminal Móvil
UART	Transmisor/Receptor Asíncrono Universal (<i>Universal Asynchronous Receiver-Transmitter</i>)
USB	<i>Universal Serial Bus (Bus serie universal)</i>
UWB	Banda ultra-amplia (<i>Ultra-Wide Band</i>)
Wi-Fi	Fidelidad inalámbrica (<i>Wireless Fidelity</i>)
WLAN	LAN inalámbrica (<i>Wireless LAN</i>)
WPAN	Red de área personal inalámbrica (<i>Wireless Personal Area Network</i>)
ZC	Coordinador ZigBee (<i>ZigBee Coordinator</i>)
ZDO	Objetos de dispositivo ZigBee (<i>ZigBee Device Objects</i>)
ZED	Dispositivo final ZigBee (<i>ZigBee End Device</i>)
ZigBee	Estándar específico para redes de sensores inalámbricas de bajo consumo
ZigBee Stack	Pila de protocolos que agrupa todos los niveles y especificaciones que añade el estándar ZigBee
ZR	Router ZigBee (<i>ZigBee Router</i>)

1. Introducción

1.1 Objetivo

El objetivo de esta tesis es diseñar y construir un sistema funcional que permita localizar y monitorear objetos en espacios cerrados (“Indoor”) empleando señales de radiofrecuencia que se basen en la tecnología ZigBee.

Los puntos principales que se persiguen son:

- Definir y diseñar una arquitectura de sistema que permita la implantación de un sistema de localización de objetos en interiores, en un entorno que constará de una única planta, y su superficie oscilará entre 50 y 300 metros cuadrados.
- Implementar todo el software necesario para el sistema de localización. Esto incluirá, entre otros, un programa con Interfaz Gráfica de Usuario (GUI) que ejecutará el usuario, cuya intención sea localizarse en una planta y localizar a otros usuarios.
- Investigación y estudio de los materiales, dispositivos y circuitos de radiofrecuencia de empleo comercial para diseño del hardware que cumplan con el estándar IEEE 802.15.4.
- Validar experimentalmente el sistema en un entorno físico.

1.2 Motivación

Los sistemas de localización en interiores han cobrado un interés creciente debido a la reducción en el tamaño y en el costo de los dispositivos inalámbricos, incrementado su uso para monitorear ambientes específicos. Pueden ser utilizados para llevar a cabo un seguimiento de bienes o personas que permita conocer sus posiciones en todo momento. El conocimiento en tiempo real de la ubicación del personal, los bienes, y/o los instrumentos portátiles repercute en un aumento de la eficiencia de los sistemas al cual pertenecen. Ahora, la estimación de la ubicación de un objeto, se refiere al proceso de obtención de información relacionada a la ubicación en un nodo, con respecto a un conjunto de posiciones de referencia conocidas.

Existen multitud de aplicaciones que demandan el conocimiento de la posición, muchas de ellas consisten en el seguimiento de personas u objetos, tanto en el exterior como en el interior de edificios, con el objetivo de proporcionar información basada en el contexto deseado. Por ejemplo, un visitante en un museo puede llevar un dispositivo de audio/vídeo que ofrece información de interés para el visitante, dependiendo de su ubicación en el museo, el sistema necesitará conocer la localización del usuario, así como su orientación, y al oprimir un botón reproducirá la información relacionada a la obra de arte que se encuentre observando la persona. Estos tipos de aplicaciones cada día más novedosas buscan mejorar la manera de vivir de las personas.

La zona de cobertura de la localización puede variar desde entornos muy reducidos, como sería un servicio de localización de personas en el interior de un hospital, hasta ámbitos muy amplios, como sería la localización de ambulancias a escala nacional. En ambos casos, el objetivo es ofrecer un servicio de localización.

Para el servicio de ámbito nacional existen multitud de sistemas que permiten cumplir con el objetivo de localizar, siendo el más importante el Sistema de Posicionamiento Global (GPS, de sus siglas en inglés Global Positioning System). Sin embargo, para el servicio en interiores es necesario desarrollar un sistema de localización apropiado que sea capaz de manejar la complejidad de la estructura propia de un espacio cerrado.

Entre las múltiples opciones existentes para desarrollar un sistema de localización *indoor*, destacan las redes de sensores que utilizan señales de radiofrecuencia, en el que, aunque cada elemento de la red posee recursos limitados a través de estrategias de comunicaciones y procesamiento cooperativo, permite a la red de forma distribuida realizar las tareas complejas.

1.3 Antecedentes

La investigación destinada a la localización en interiores se origina tras la imposibilidad de los sistemas GPS de localizar dentro de un edificio de manera adecuada. La localización en interiores tiene un futuro prometedor en el ámbito de la *“Ubiquitous Computing”*; disciplina en la que se pretende que una persona utilice en su vida cotidiana múltiples dispositivos electrónicos, pero que no tenga la necesidad de darse cuenta de que lo está haciendo. Por ejemplo; supongamos que un doctor de un hospital está visitando a un paciente, y necesita consultar una copia impresa de la historia clínica de éste, así que desde la habitación, selecciona en su PDA o algún otro dispositivo *“Imprimir historia clínica”*, y el sistema localiza al médico en la habitación, consulta en la base de datos qué paciente está ingresado en dicha habitación, y envía una orden de impresión del expediente de dicho paciente en la impresora más cercana a la posición del doctor. Del mismo modo, lo anterior se podría extrapolar a otros entornos; como por ejemplo almacenes, restaurantes, hoteles y grandes empresas, entre otros.

Estos sistemas son conocidos como Sistemas de Localización; los cuales, por lo general usan pequeños transmisores de bajo consumo de energía llamados etiquetas o *tags*, colocados sobre los objetos o llevados por las personas a localizar, y un juego de lectores trazan su posición.

La tecnología inalámbrica está basada en la transmisión de ondas electromagnéticas, que son emitidas a través de antenas hacia todas las direcciones. Los avances en las comunicaciones inalámbricas y en la electrónica, han hecho viable el desarrollo de redes multifuncionales de bajo costo y bajo consumo energético, motivo por el cual su uso se extiende a muchos sectores para los cuales se han desarrollado protocolos capaces de satisfacer las necesidades particulares de éstos.

El nacimiento de las WLAN (Wireless LAN) se remonta hacia el año de 1979, según los resultados de un experimento realizado por ingenieros de IBM en Suiza, que tenía como fin el utilizar enlaces infrarrojos para la creación de una red local de una fábrica. Las investigaciones siguieron adelante, tanto en el rango del infrarrojo como en el de las microondas, donde se utilizaba el esquema del espectro esparcido (spread-spectrum), siempre a nivel de laboratorio. En mayo de 1985, y tras cuatro años de estudios, la FCC (Federal Communications Commission), agencia federal del Gobierno de Estados Unidos encargada de regular y administrar en materia de telecomunicaciones, asignó las bandas IMS (Industrial, Scientific and Medical) 902-928 [MHz], 2,400-2,4835 [GHz], 5,725-5,850 [GHz] a las redes inalámbricas basadas en espectro esparcido. IMS es una banda para uso comercial sin licencia, es decir, la FCC simplemente asigna la banda y establece las directrices de utilización, pero no se involucra ni decide sobre quién debe transmitir en esa banda.

La asignación de una banda de frecuencias propició una mayor actividad en el seno de la industria, ese respaldo hizo que las redes WLAN dejaran de ser utilizadas solo en laboratorios e iniciaron sus aplicaciones fuera de estos.

La utilización de la tecnología inalámbrica al máximo permite obtener y comunicar información desde y hacia cualquier dispositivo móvil de comunicaciones, como es el caso de AP de internet o celulares, entre otros.

Los campos de aplicación de las tecnologías inalámbricas son innumerables, algunas de estas aplicaciones son: en telemática, agricultura de precisión, exploración de hábitats, detección y actuación ante desastres, edificios inteligentes, gestión de centros gubernamentales, monitorización sanitaria y médica, entre otras. La cuestión crítica en la mayoría de las aplicaciones es relacionar los datos físicos medidos con la posición en el espacio. Esta cuestión se conoce con el nombre de localización, es decir, para cada dispositivo se debe conocer su posición exacta en el espacio, en cualquier instante de tiempo.

1.4 Organización del trabajo

Este trabajo está estructurado en ocho capítulos, según se indica a continuación:

- En el Capítulo 1 se presenta el objetivo y la estructura del presente Proyecto.
- En el Capítulo 2 se realiza una revisión de los sistemas existentes de localización de objetos en exteriores e interiores.
 - o En primer lugar se revisan las distintas tecnologías disponibles actualmente para estimar la posición.
 - o Posteriormente se presentan las distintas técnicas de localización posibles.
- En el Capítulo 3 se detallan las características del RSSI, así como las técnicas existentes para estimar la localización en base a las características de la señal de RF.
- En el Capítulo 4 se analiza el estándar 802.15.4. en el que está basado el Stack ZigBee, tanto sus características técnicas como el tipo de aplicaciones para el que está orientado.
- En el Capítulo 5 se introduce al diseño del proyecto, incluyendo los diseños electrónicos para los distintos dispositivos, así como su configuración y programación.
- En el Capítulo 6 se detallarán los aspectos relacionados al lenguaje de programación utilizado, así como una descripción del funcionamiento de la aplicación.
- En el Capítulo 7 se describen los distintos experimentos realizados durante el desarrollo del Proyecto y los resultados obtenidos.
- En el Capítulo 8 se presentan las conclusiones y líneas futuras del Trabajo.

Por último, se incluye la bibliografía utilizada para el desarrollo de este Proyecto.

2. Sistemas de localización de objetos

Para iniciar el capítulo se incluye la definición general de un sistema de radio localización:

Un sistema de radio localización es un conjunto de tecnologías, equipamientos físicos y recursos humanos destinados al rastreo, monitoreo y control, para ofrecer información a las personas (o máquinas), en todo momento y con precisión, para determinar la posición de personas u objetos.

Para conocer la posición de un objeto en un mapa, área o espacio determinado, se suele utilizar un método basado en medir la distancia entre dicho objeto y una de las referencias. Para medir la distancia entre dos puntos de forma inalámbrica existen diferentes formas de hacerlo, a continuación se presentan algunos de los Sistemas existentes para dicho propósito.

2.1 Sistemas de localización en espacios abiertos

2.1.1 Sistema de Posicionamiento Global

El Sistema de Posicionamiento Global (Global Positioning System GPS) es un sistema de radiolocalización que, además de la posición del dispositivo, también permite conocer su velocidad del movimiento, la orientación del desplazamiento y la traza del recorrido que se ha efectuado.

El GPS, emiten señales de radio en forma ininterrumpida por un conjunto de satélites que orbitan a más de 20,000 [km] de altitud. Estas señales contienen datos relativos a la posición del satélite en el espacio y a la hora actual en un formato internacional denominado UTC (Universal Time Coordinated).

Para el buen funcionamiento del sistema, cada satélite debe mantener una órbita extremadamente precisa, estable y debe ser “visible” desde la superficie terrestre. Visible, en este contexto, no significa que se pueda distinguir a simple vista, sino que las señales que emite puedan ser recibidas “en línea recta”, es decir, que nada se interponga entre el satélite y el receptor GPS.

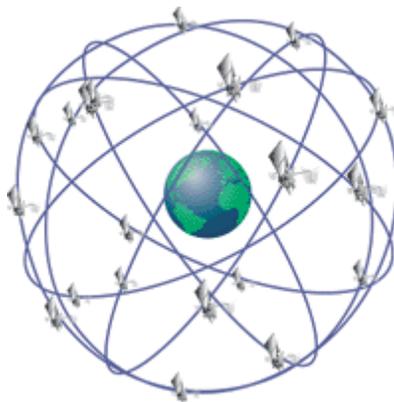


Figura 2.1. Esquema de la localización por GPS

Un esquema básico del GPS se muestra en la Figura 2.1; este sistema tiene una precisión que puede llegar a ser menor de 10 metros si se toman en consideración

más de cuatro satélites, lo cual es bastante interesante para realizar una localización en el mundo a nivel global, pero no tiene mucho sentido usarlo para localización dentro de un área pequeña, como puede ser un edificio.

Las señales del GPS viajan muchos kilómetros y son bastante tenues, por lo que un receptor GPS en el interior de un edificio no es muy adecuado para encontrar señales procedentes de los satélites; y más aún, para conseguir que estas señales le sirvan para ser localizado. Además, el GPS, como su nombre indica, es un sistema de posicionamiento global, y no está por tanto indicado para permitir la localización a nivel local.

2.1.2 RADAR (Radio Detection And Ranging)

El radar (término derivado del acrónimo inglés Radio Detection And Ranging, “detección y medición de distancias por radio”) es un sistema electrónico que permite detectar objetos y determinar su distancia. El principio de funcionamiento de este sistema es la transmisión de una determinada señal de Radiofrecuencia que incide en un objeto llamado "blanco", el cual refleja la señal en varias direcciones, una porción de esta señal "eco" es captada por un receptor, que puede ser la misma antena de transmisión, la cual se encarga de filtrar la señal de un cierto ruido "clutter", amplificarla y procesarla para obtener la información del "blanco".

Al medir el tiempo entre la señal transmitida y la recibida, así como por la posición de la antena, en elevación y azimut, se puede determinar la posición del "blanco".

2.1.3 Localización usando telefonía móvil

Otra alternativa posible, que además no necesitaría ningún hardware adicional, es el uso de un teléfono móvil, y de hecho ya hay operadoras de telefonía móvil que ofrecen la opción de localización vía móvil a sus abonados. Sin embargo, la falta de precisión sitúa a esta tecnología en clara desventaja respecto de otras, ya que los sistemas de localización de este tipo no pueden dar precisiones mayores de 50 metros, por lo que no funcionaría adecuadamente.

Lo Anterior es debido a que la localización con el uso del teléfono móvil (localización por GSM) se basa en la detección de la célula a la que se encuentra conectado el teléfono móvil. En zonas urbanas la precisión es de decenas de metros, sin embargo, en las zonas rurales, donde se necesitan menos células para dar servicio a menor población, esta precisión es mucho menor. Por lo que, ésta es una clara desventaja de la tecnología GSM que hace totalmente inapropiado su uso para la localización en interiores.

2.1.4 GPS Asistido (Assisted Global Positioning System)

Un dispositivo móvil típicamente opera en conjunto con un servidor de localización. El dispositivo y el servidor de localización comparten información de posicionamiento a través de una red de telecomunicaciones.

Cuando un dispositivo móvil realiza una petición de localización, la red inalámbrica envía la localización más aproximada del dispositivo móvil al servidor de localización, entonces el servidor de localización le envía al dispositivo móvil la información de qué satélites deberían ser relevantes para calcular su posición. Con esta información el dispositivo móvil lee la señal de los satélites apropiados, calcula su distancia con todos los satélites a la vista y envía esta información de regreso al servidor de localización. Este sistema comete un error alto, entre 5 y 50 [m]. Compañías como Atmel y U-blox han diseñado receptores con una alta

sensibilidad que permiten detectar señales GPS incluso en el interior de los edificios, pero estos sistemas también proporcionan poca precisión en sus estimaciones. En la Figura 2.2 se puede apreciar el esquema general del sistema.

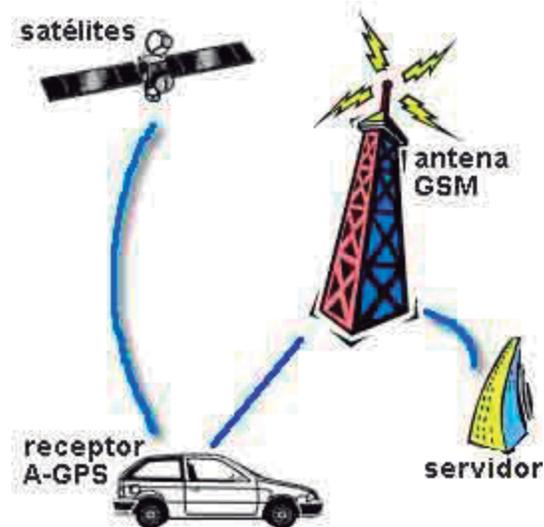


Figura 2.2. Sistema A-GPS

2.2 Sistemas de localización en espacios cerrados

Se presenta una definición de los sistemas de localización en espacios cerrados, o mejor dicho, de los Sistemas de Radiolocalización en interiores (SRLI) con el fin de introducir los elementos básicos que conforman ha dicho sistema:

Un SRLI es un sistema cuyo objetivo es proporcionar información sobre la localización de un objeto situado en el interior de un edificio, utilizando información extraída del campo electromagnético propagado entre el objeto y una infraestructura de radiocomunicaciones compuesta de un conjunto de terminales fijas.

Un elemento importante de un SRLI es el escenario en el que se desarrolla el servicio de localización, que denominamos zona de cobertura del sistema, y que en este caso es el interior de un edificio. Este elemento condiciona el modo en el que el campo electromagnético se propaga, haciendo difícil la predicción del comportamiento del mismo en función de la posición.

A continuación describiré las principales tecnologías existentes que permiten hacer estimaciones de localizaciones en interiores.

2.2.1 Identificación por Radiofrecuencia RFID

RFID (Radio Frequency IDentification) sistema que se basa en etiquetas de radiofrecuencia que contienen una antena emisora/receptora que al ser excitada por un transmisor emite una señal. Así, un usuario que se quisiera localizar en un edificio tendría cerca de él un número de etiquetas de radiofrecuencia. El propio usuario tendría un lector de etiquetas RFID y al obtener información de las etiquetas cercanas puede llegar a localizarse.

Un ejemplo de un sistema de localización que usa RFID es *Cricket*, un sistema ideado por ingenieros del MIT (Massachusetts Institute of Technology), cuya precisión es 2 centímetros y ha sido empleado en otros proyectos como seguimiento de objetos, control de robots o en aplicaciones *context-aware*¹ (en las cuales la localización del usuario juega un papel muy importante). Este método parece correcto y puede localizar de manera satisfactoria, pero sin embargo es aún un método costoso, por esta razón se buscaron otros métodos de localización de menor costo.

¹ Término de la ubiquitous computing, que se utiliza para los dispositivos que tienen información sobre las circunstancias en las que operan y pueden reaccionar en consecuencia. Dispositivos sensibles al contexto que también puede tratar de hacer suposiciones acerca de la situación actual del usuario.

Como resumen, la localización por RFID es una alternativa válida, aunque es muy cara por la gran cantidad de lectores RFID que se ocupan para una localización correcta, lo anterior se debe a que el alcance de estas señales es muy reducido.

2.2.2 Localización por Infrarrojos

La localización por infrarrojos se puede desechar en un primer momento para la localización en interiores, por ser de corto alcance (unos dos metros) y porque, además, se requieren enlaces LoS (Line of sight). Por su corto alcance habría que incluir una cantidad enorme de emisores de infrarrojos, y aún así serían imposibles de detectar ciertas localizaciones por el problema de LoS.

2.2.3 Bluetooth

La tecnología Bluetooth es una especificación abierta para la comunicación inalámbrica de datos y voz. Está basada en un enlace de radio de bajo costo y corto alcance, implementado en un circuito integrado, proporcionando conexiones instantáneas para entornos de comunicaciones tanto móviles como estáticas.

La ventaja es que es una tecnología barata, pero el alcance es demasiado corto y se necesitarían muchos dispositivos para cubrir una planta. El error cometido puede estar en torno a 1,5 metros, lo cual no está mal para un interior. El mayor inconveniente que tiene Bluetooth es que el indicador de potencia recibida RSSI no es preciso, por lo que no se puede usar y por ello, si se encuentra un dispositivo cercano, hay que asumir que se está en su entorno, pero no se puede estimar el grado de cercanía o lejanía.

2.2.4 Wi-Fi (estándar IEEE 802.11)

Wi-Fi, al igual que las otras tecnologías que estamos analizando, consiste en un sistema de envío de datos sobre redes que utilizan ondas electromagnéticas en lugar de cables. Se basa en el estándar IEEE 802.11, y es por ello que podemos encontrar diversos tipos de Wi-Fi:

- Los estándares IEEE 802.11 son fácilmente aceptados debido a que usan la banda frecuencia de 2.4GHz.
- La tasa de transferencia del medio depende del estándar que se haga uso. Puede proporcionarnos una tasa de transmisión de 11 [Mbps] (IEEE 802.11b) hasta 108 [Mbps] (IEEE 802.11n), pasando por los 54 [Mbps] del estándar IEEE 802.11g.
- En la actualidad ya se utiliza el estándar IEEE 802.11a, conocido como Wi-Fi5, ya que opera en la banda de 5[GHz]. Dicha banda de frecuencia (5[GHz]) ha sido recientemente habilitada. Al no existir otras tecnologías como ZigBee o Bluetooth que hagan uso de ella, garantiza un mínimo de interferencias vecinas. Su alcance es algo menor que el de los estándares que trabajan a 2.4GHz.

La principal ventaja que proporciona Wi-Fi, aparte de las altas tasas de transmisión, es la capacidad de suministrar cobertura en un gran rango de distancia (capaz de alcanzar los 100 metros).

Por tanto, resulta atractiva la posibilidad de utilizar esta tecnología, ya que los costos de instalación se reducen drásticamente respecto al resto de posibilidades mencionadas. La precisión de los sistemas comerciales actuales oscila entre 3 y 30 [m], y todos ellos utilizan la potencia recibida como característica esencial del campo electromagnético.

Por otra parte, la desventaja fundamental de estas redes existe en el campo de la seguridad. Existen algunos programas capaces de capturar paquetes enviados y calcular la contraseña de la red, y de esta forma acceder a ella.

Vistos los principales puntos que nos puede aportar la utilización de Wi-Fi (altas tasas de transmisión y gran cobertura), podemos concluir que este tipo de medio quedaría fuera del tipo de transmisión que queremos llevar a cabo al necesitar una mayor fuente de energía para poder proporcionar dichas tasas de transmisión.

2.2.5 UWB (Ultra Wideband)

La tecnología UWB parece una buena opción para la localización en interiores, ya que permite altas tasas de transmisión; y gracias a la corta duración de sus pulsos y a la gran resolución temporal de los receptores UWB, es posible hacer medidas muy precisas del tiempo que tarda la señal en llegar al receptor, permitiendo errores de menos de 20 cm mediante triangulación. Pero dicha precisión es solo alcanzada en un metro y el otro inconveniente es que no está regulado de tal forma que se pueda usar libremente a escala masiva.

2.2.6 ZigBee

ZigBee es el nombre de la especificación de un conjunto de protocolos de alto nivel de comunicación inalámbrica, basada en el estándar IEEE 802.15.4. Define el nivel físico y el control de acceso al medio de redes inalámbricas de área personal (PAN) con bajas tasas de transmisión de datos. Las principales aplicaciones de la redes ZigBee se centran en redes de sensores en entornos industriales, médicos y principalmente domóticas. Dentro de sus ventajas principales se tienen su bajo costo así como una relativa sencillez en el despliegue de la red, en contraste se tienen sus desventajas, algunas de las cuales son que los dispositivos con los que se trabaja suelen contar con capacidades limitadas en poder de cómputo así como las posibles interferencias electromagnéticas con otros dispositivos que operen en la misma banda de frecuencia.

2.2.7 Elección de tecnología inalámbrica

De los distintos sistemas inalámbricos analizados se puede llegar a concluir que cada uno de estos se adapta mejor a un tipo de necesidades. Para la aplicación de este trabajo son primordiales el bajo consumo y el acceso al RSSI, por lo tanto la tecnología que mejor se adapta es la ZigBee. Esta tecnología se describirá en capítulos posteriores, para dar una mayor cantidad de datos y de información.

Una vez que se han expuesto las posibles tecnologías que permiten llevar a cabo la localización en interiores, se describen las razones por las cuales, frente a todas estas alternativas, se eligió la transmisión inalámbrica siguiendo el protocolo 802.15.4:

- No tiene un alcance muy elevado, lo que sería contraproducente para un sistema de localización que quiere discriminar entre localizaciones situadas a, como mucho, decenas de metros una de la otra.
- Puede atravesar con relativa facilidad las paredes, por lo que no es necesario el uso de innumerables dispositivos para cubrir totalmente una área de 200 a 300 metros cuadrados.
- La tecnología utilizada actualmente en aplicaciones que requieren comunicaciones seguras con baja tasa de envío de datos y maximización de la vida útil de sus baterías.
- Para el funcionamiento del sistema de localización, no es necesario agregar ningún hardware especializado adicional a los dispositivos.

2.3 Técnicas de estimación de distancias

El posicionamiento inalámbrico involucra la transmisión de una señal inalámbrica, y en la recepción, alguna propiedad de la señal recibida se usa para determinar las posiciones relativas del receptor o transmisor.

Las características medidas de la señal normalmente usadas para posicionar son: la intensidad de señal recibida (RSS); el tiempo de recepción de señal o tiempo de llegada (ToA) y su variación, que es la diferencia de tiempo de llegada (TDoA); y la dirección en la que se recibe la señal ó ángulo de llegada (AoA).

A continuación se detallan las técnicas de localización más usadas para permitir la estimación de distancias, y se explicará cuál se ha elegido en este proyecto y por qué.

2.3.1 Tiempo de llegada (Time of Arrival ToA)

Esta técnica mide el tiempo que tarda una señal en viajar de un nodo a otro a una velocidad conocida. El retardo de propagación es la cantidad de tiempo requerido por una señal para viajar desde el transmisor hacia el receptor, como se aprecia en la Figura 2.3., y puede calcularse como $t_1 - t_0$.

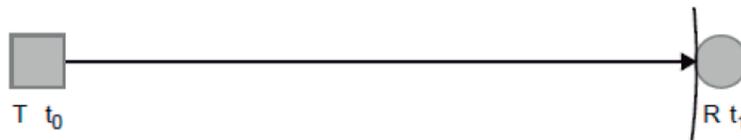


Figura 2.3. Tiempo de llegada.

Multiplicando el tiempo de propagación $t_1 - t_0$ por la velocidad de propagación de la señal (aproximadamente la velocidad de la luz $c \approx 300,000$ [Km/s]), se obtiene la distancia entre el transmisor y el receptor.

Esta técnica es afectada por 2 fuentes de error: efectos multitrayecto y condiciones de trayectoria directa no detectada. Los efectos multitrayecto resultan en reflexiones y retransmisiones de trayectorias que se reciben a lo largo de la trayectoria directa, causando errores.

Para tener mediciones razonablemente confiables de los tiempos de propagación, los relojes del *tag*² y de los lectores³ deben estar sincronizados. Para alcanzar una precisión en una escala de nanosegundos (1×10^{-9} [s]), lo que resulta en medición más precisa de la distancia, se debe realizar un sistema de sincronización de reloj muy sofisticado, lo que implica altos costos en términos de desarrollo y esfuerzo.

2.3.2 Diferencia de tiempos de llegada (Time Difference of Arrival TDoA)

Aunque la técnica anterior es muy precisa, se necesita una estricta sincronización entre el emisor y el receptor. La técnica llamada diferencia de tiempos de llegada (TDoA) permite una sincronización en el tiempo más compleja. El transmisor emite un pulso acústico (ultrasonido) junto con un pulso de radio, mientras que el receptor compara el tiempo de llegada de ambos pulsos. Conociendo la velocidad de cada una de las señales, la diferencia de tiempo entre las llegadas revela la distancia entre el emisor y el receptor.

² Pequeños nodos autónomos denominados "motes" o "tags", están integrados por un microcontrolador y un transmisor/receptor; que permiten centralizar información.

³ Aparatos electrónicos que captan la información enviada por los tags.

Aunque estos sistemas ofrecen errores muy pequeños en la estimación, tienen dos limitaciones que reducen notablemente su aplicabilidad en el mundo real. La primera es su limitada cobertura (entre 3 y 15 metros), lo que supone sólo una fracción del rango de comunicación de los transmisores de radiofrecuencia. La segunda es que requieren un par emisor/receptor por separado, lo que implica mayor tamaño, costo y consumo de energía. La Figura 2.4 muestra lo antes descrito:

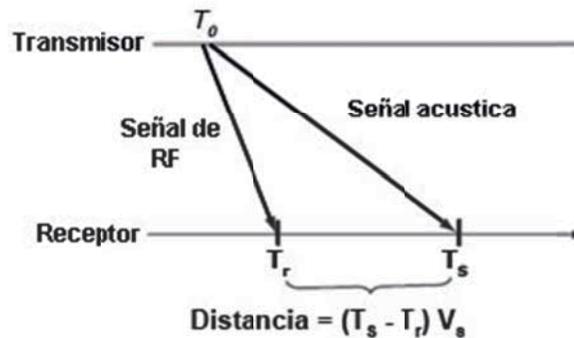


Figura 2.4. Estimación de la distancia por la diferencia de tiempos de llegada.

Otro inconveniente para nuestro caso es que no funciona en interiores, y es debido al multirayecto, ya que el tiempo de propagación será más elevado de lo que sería en espacio libre y por lo tanto las mediciones de distancias no serán muy precisas. También requiere de un reloj muy preciso para poder medir con variaciones temporales muy pequeñas.

2.3.3 Ángulo de llegada (Angle of Arrival AoA)

El ángulo de llegada se define como el ángulo que forma la dirección de propagación de una onda incidente con una determinada dirección de referencia, se despliegan varias antenas de forma que pueda inferirse el ángulo de llegada. Esta técnica es efectiva cuando no hay multirayectos ya que en caso contrario podría tomarse el ángulo de una señal reflejada.

Este método se ilustra en la Figura 2.5, donde R_1 es el lector y T es el tag. Utilizando las posiciones de dos lectores en ubicaciones conocidas, la posición del *tag* puede determinarse mediante simple triangulación. Para cada lector, el ángulo de llegada de la señal recibida desde el mismo *tag* es calculado, y entonces un algoritmo es utilizado por el sistema de localización para determinar la posición del *tag*.

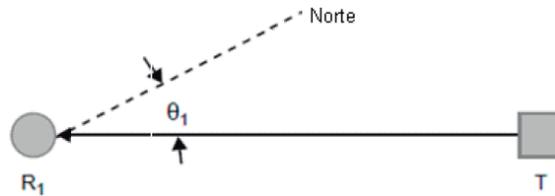


Figura 2.5. Método de Ángulo de Llegada.

Este método puede visualizarse en la Figura 2.6, donde dos lectores R_1 y R_2 son usados y donde T , una vez más, denota el *tag* cuya posición está siendo determinada. N indica el norte.

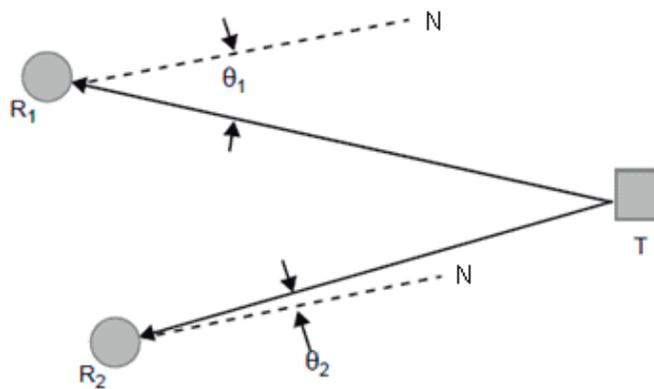


Figura 2.6. Estimación de la distancia por el ángulo de llegada.

Para poder tomar mediciones mediante este método se requiere un juego complejo de entre 4 y 12 antenas situadas en línea horizontal en varias ubicaciones.

La exactitud del sistema aumentará conforme se incrementa el número de arreglos de antenas utilizados. Además del costo que esto implica, las mediciones de los ángulos son muy sensibles a la interferencia por multitrayectoria, común en ambientes cerrados, por lo que esta técnica funciona mejor en lugares donde existe línea de vista entre lectores y *tags*.

2.3.4 Tiempo de Vuelo (Time of Flight ToF)

Este método usa mediciones del tiempo transcurrido en una transmisión entre un *tag* y un lector, basado en la velocidad de propagación estimada de una señal típica a través del medio. Ya que está basado en un valor de tiempo, la exactitud del reloj es mucho más importante que en los métodos anteriores.

Lectores R , con relojes de alta exactitud transmiten señales con valores de tiempo de salida conocidos hacia los *tags* T u otros lectores, también con este tipo de relojes. El tiempo de salida t_1 es comparado con el tiempo de llegada t_2 , y realizando una estimación de la velocidad de propagación de la señal S , se puede determinar la distancia D entre los dispositivos con una exactitud dentro de 1 ó 2 metros.

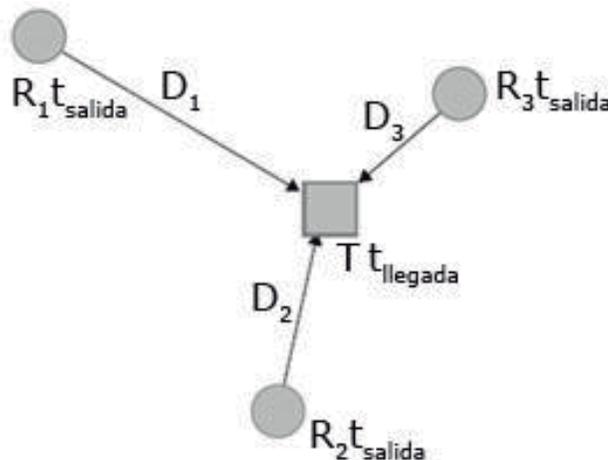


Figura 2.7. Forma en la que se determina la posición del tag con ToF.

Este método no necesita agregar hardware adicional al sistema, así que se puede utilizar el mismo hardware utilizado para comunicación de datos y procesamiento de señales.

Un sistema ToF ideal requiere del uso de costosos relojes de alta exactitud; además la señal también es afectada por la interferencia de otras señales, ruido y la propagación multitrayectoria. Tiene una exactitud relativamente más alta, comparado con los otros métodos, al mismo tiempo es considerado como un método seguro para RTLS (Real-Time Locating Systems).

2.3.5 Indicador de Intensidad de Señal Recibida (Received Signal Strength Indication RSSI)

Esta técnica utiliza las propiedades de atenuación de la señal de radio para modelar la distancia entre dos nodos. Este método utiliza varios puntos de acceso WLAN (Wireless Local Area Network) simultáneamente para obtener la localización de un dispositivo. Las intensidades de las señales recibidas, de al menos tres dispositivos, son utilizadas para determinar la posición del objeto o persona a localizar.

Es el método más utilizado en redes tipo WLAN para localización en interiores o ambientes cerrados, como oficinas o departamentos, debido a que no requiere ningún hardware adicional, ya que la mayoría de dispositivos WLAN proveen la información de las mediciones RSSI.

Los métodos basados en la intensidad de señal recibida son ampliamente divididos en dos grupos:

- *Huella digital de radiofrecuencia*
- *Modelo de propagación.*

La huella digital de radiofrecuencia o *fingerprinting* consiste en generar un mapa de intensidad de la señal recibida en el área de interés, anotando el valor de intensidad de la señal por cada punto de acceso en cada posición en el mapa. La distancia entre cada punto de medición influye en la exactitud, por lo que se debe seleccionar un espaciamiento coherente, sin llegar a al extremo de tener una rejilla muy fina, ni puntos muy lejanos entre sí.

Para determinar la posición de un objeto o persona en particular, se lee la intensidad de la señal recibida de todos los puntos de acceso, y usando un algoritmo se encuentra el punto más cercano que mejor coincide con los valores de lectura de la señal que se midieron en el levantamiento inicial y que se tienen almacenados en una base de datos.

Una de las desventajas de este método es que al realizar el levantamiento de las mediciones en los puntos, se realiza con condiciones iniciales, las cuales pueden cambiar según cambie el entorno; por ejemplo al agregar una pared o al modificarse el tamaño de los muebles, se afecta los valores medidos anteriormente debido a la pérdida por multitrayecto, y entonces pueden no coincidir los valores medidos en un inicio con los actuales, lo cual provoca errores en la localización; este problema se resolvería volviendo a realizar las mediciones del levantamiento.

Por otro lado, en los sistemas que utilizan un modelo de propagación, la distancia entre un *tag* (objeto o persona) y un lector está determinada mediante la conversión del valor de intensidad de la señal en el lector, en una medición de distancia basada en la potencia de salida, conocida de la señal del tag y en un modelo particular de pérdida de trayectoria. Un servidor de localización, utiliza un algoritmo para estimar la localización del *tag* utilizando las distancias calculadas entre el *tag* y los lectores.

Este método es bastante complejo, y también está sujeto al problema de cambios en el entorno, que se deben actualizar en el modelo de propagación para cada

trayectoria. Aunque la determinación de una distancia en particular entre un *tag* y un lector difiere sustancialmente del método ToA, el cálculo de la ubicación consiste en un algoritmo similar.

El método RSSI por modelo de propagación se ilustra en la Figura 2.8., donde T denota el *tag*, R_1 , R_2 , R_3 son los lectores, y la intensidad de la señal de cada lector está denotada por S_1 , S_2 , y S_3 respectivamente.

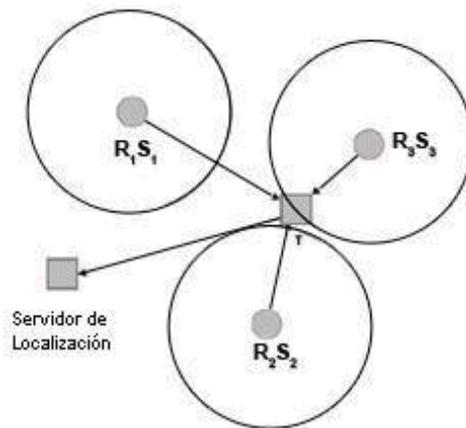


Figura 2.8. Estimación de la ubicación con RSSI.

Para ser efectivo, RSSI requiere un despliegue denso de dispositivos (lectores), lo cual incrementa considerablemente el costo del sistema. Sin embargo, el problema clave con los sistemas basados en RSSI es que un modelo de pérdida por trayectoria adecuado debe ser encontrado para entornos tanto sin línea de vista como para no puntos estacionarios.

2.3.6 Elección de la técnica de localización

Con base a las características y funcionamiento de las distintas técnicas de localización, y a que el sistema de posicionamiento debe cumplir con los requerimientos de precisión, eficiencia en el consumo de energía, escalabilidad y costo; el método más conveniente es el basado en RSSI, motivado por el bajo

consumo de energía de los dispositivos, bajo costo al no requerir de hardware adicional, ya que en caso de haber elegido hacer una localización tipo ToA se tendría que dotar de un reloj muy preciso a los dispositivos, y en el caso de AoA se debería disponer de muchas antenas en cada transmisor, además que, como se indicó, la mayoría de dispositivos WLAN proveen de información de RSSI sin requerir alguna modificación a el hardware.

De los dos grupos de métodos RSSI expuestos, el que será utilizado es el de huella digital de radiofrecuencia o *fingerprinting*, por la flexibilidad, facilidad de calibración y actualización de la información de referencia, además de que para utilizar el método por propagación es muy difícil encontrar un modelo que se ajuste a la realidad de las edificaciones, tomando en cuenta sobretodo el ambiente cambiante y dinámico de los entornos interiores.

Cabe señalar que los valores de RSSI son muy variantes en estos entornos, dado que su valor es en función de los distintos tipos de antenas utilizadas, fabricantes, o incluso del nivel de carga de la batería del dispositivo, y el desarrollo del trabajo no está enfocado a determinar la calidad y precisión de los indicadores de señal recibida, sino de relacionar esta propiedad a la distancia para un ambiente interior dado. La hipótesis es que este método dé resultados aceptables.

3. Indicador de Potencia Recibida RSSI

Los sistemas aplicados con base a ultrasonidos o infrarrojos normalmente se dedican sólo a la determinación de la ubicación estimada de los objetos y en cambio en un sistema basado en RF, la estimación de posición de los objetos puede ser sólo una característica añadida al sistema existente, y la red inalámbrica puede utilizarse para otros tipos de actividades tales como la comunicación de datos y control.

La elección del algoritmo de estimación de la localización depende del escenario, los métodos de localización-estimación son comparados en función de su rendimiento y complejidad. La exactitud en la localización, que es la distancia entre la ubicación real y la ubicación estimada, es la más intuitiva de las métricas de desempeño. Esta es la mayor diferencia entre la implementación de un sistema de localización con una precisión de centímetros frente a un sistema donde un error de estimación de localización es de hasta varios metros, que sigue siendo aceptable.

Sí el objetivo de estimar la ubicación es realizar un seguimiento del personal en un edificio, una precisión de varios metros puede ser suficiente, por lo que la simplicidad y el bajo costo en la implementación de sistemas de localización llevada a cabo, sobre una base de protocolos de corto alcance inalámbrico, es más que suficiente.

La estimación de localización por lo general consta de dos grupos de nodos. El primer grupo se compone de nodos fijos con lugares conocidos. Estos nodos fijos, se denominan nodos de anclaje o nodos baliza, los cuales se utilizan como referencia para la estimación de la ubicación. El segundo grupo de nodos, los cuales se encuentran en lugares desconocidos, serán los nodos de seguimiento. Por lo que el objetivo principal será la estimación de la ubicación de los nodos de seguimiento con la ayuda de los nodos baliza.

La idea básica de posicionamiento local puede resumirse con lo siguiente: Un nodo de seguimiento con destino desconocido emite una señal que es recibida por los nodos de anclaje vecinos. El nodo ancla mide la intensidad de la señal recibida (RSSI), valores que son utilizados como fuente de información para un algoritmo que determina la ubicación aproximada del nodo de seguimiento.

3.1 Características y especificaciones

Existen diferentes formas de medir la potencia de una señal de radiofrecuencia recibida por un dispositivo. Desde un punto de vista físico, dicha potencia puede ser medida en Watts [W] o miliwatts [mW] o, en valores logarítmicos, en [dBW] (potencia logarítmica referida a 1 [W]) o dBm (potencia logarítmica referida a 1 [mW]). De entre todas ellas, para medir la potencia recibida en un dispositivo resulta más conveniente la medida en unidades logarítmicas, debido al amplio rango de valores que la potencia puede tomar. De las dos unidades logarítmicas mencionadas, es habitual en comunicaciones inalámbricas utilizar el [dBm], debido a que la potencia recibida puede presentar valores muy bajos. Por ejemplo, los valores típicos para dicha potencia están comprendidos entre -90 [dBm] y 20 [dBm].

En cuanto a las características de la señal podemos citar:

- Atenuación por distancia: A mayor distancia, la potencia de la señal decrece en forma logarítmica.
- Atenuaciones adicionales: Si aparecen obstáculos entre emisor y receptor, la atenuación es mucho mayor que en espacio libre, cuando una onda se topa con un obstáculo, éste absorbe parte de su energía.
- Reflexión: Una señal que avanza por el espacio libre, al toparse con un obstáculo pierde parte de su energía por la absorción, pero además, si el obstáculo es bastante grande en comparación con la longitud de onda de la señal, ésta se refleja del mismo modo que lo hace la luz al llegar a un espejo (Figura 3.1). La señal reflejada puede tener distinta fase que la señal original, dependiendo de las características del obstáculo.
- Difracción: Cuando la señal se encuentra con el borde de un obstáculo (como sucede en la esquina de una habitación), se forman frentes de onda secundarios en todas las direcciones (Figura 3.2). Que haya mayor o menor difracción depende principalmente del material del obstáculo o muro, así como de la amplitud o fase de la onda.
- Dispersión (*scattering*): La dispersión está causada por objetos que tienen dimensiones pequeñas comparadas con la longitud de onda (en contraposición con lo sucedido con la reflexión). Éstos causan también multitud de frentes de onda secundarios que se vienen a unir a todos los anteriores (Figura 3.3).

Los tres fenómenos anteriores (reflexión, difracción y dispersión) son los responsables del multitrayecto, esto es, entre la antena transmisora y la antena receptora nunca va a haber un único camino por el que transita la señal. Todos los frentes de onda que se forman se suman o restan vectorialmente unos con otros, por lo que la señal percibida en el receptor no sólo proviene del camino más inmediato (la línea recta), ni siquiera en el caso de que el emisor y el receptor se vean el uno al otro (LoS, Line of Sight). Es posible que haya una zona donde no hay señal en absoluto y a los pocos centímetros la potencia recibida sea muy grande; todo esto es debido al multitrayecto.

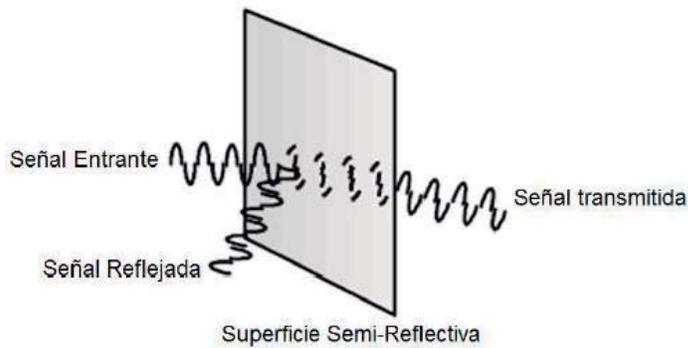


Figura 3.1. Reflexión.

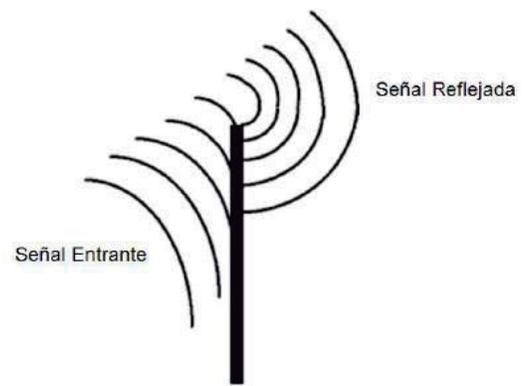


Figura 3.2. Difracción.

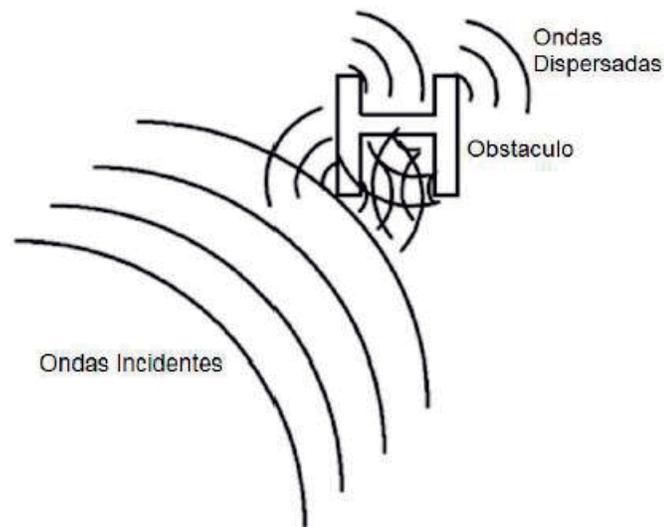


Figura 3.3. Dispersión de una onda.

La planeación del sistema de radio localización debe tomar en cuenta las siguientes cuestiones:

- Ha de asegurarse la cobertura en toda la zona de interés, por lo que es necesario hacer una primera predicción de la señal recibida en la mayoría de los emplazamientos con objeto de garantizar un mínimo de señal recibida en ambos sentidos.
- La existencia de múltiples trayectos (multipath) entre transmisor y receptor, por la existencia de múltiples obstáculos físicos entre ellos, origina réplicas de las señales originales que son causadas por los fenómenos de reflexión, difracción y refracción.

- A estas dos cuestiones hay que añadir el hecho intrínseco de que la terminal móvil se mueve y que el entorno también puede ser variable (debido a la presencia de personas y/o modificaciones en el mobiliario), por lo que el nivel de señal recibido por la estación base y móvil varían de forma continua.

3.2 Métodos de estimación de localización basados en RSSI

En entornos Indoor, el número de componentes que afectan a la señal “principal” es mucho mayor que en otros entornos, originando una mayor variabilidad de la señal. Pero puesto que la localización basada en las lecturas de potencia de la señal recibida (RSSI) es factible, se han desarrollado algunos prototipos de sistemas de localización que usan dos tipos de técnicas diferentes: localización por *triangulación* y localización por *reconocimiento de patrones*.

3.2.1 Triangulación

Los métodos de triangulación funcionan de forma similar a los mencionados ToA y DToA, ya que asimilan potencia recibida (tiempo, en el caso de ToA) a distancia. Sin embargo, al no ser precisa esta relación en interiores, se obtienen unos métodos poco precisos.

La localización se basa en la suposición de que las antenas generan un patrón de radiación en forma de esfera con centro en la propia antena. Con tres de estas esferas, cada una de ellas con un radio estimado en base a la potencia recibida de cada una de estas tres antenas, se obtendrá la estimación de la localización al intersecar las tres esferas (Figura 3.4).

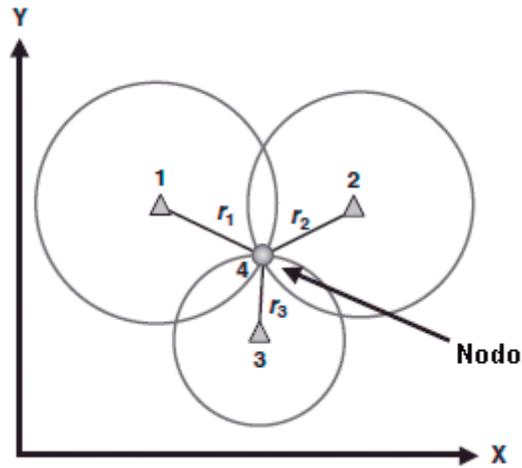


Figura 3.4. Estimación de la localización utilizando triangulación.

Esto parte de la suposición de que se haga uso de una antena con patrón de radiación esférico. Sin embargo, las antenas reales no pueden generar un patrón de radiación esférico, y por ello se usan antenas omnidireccionales, que tienen un patrón de radiación en forma de dona (*donut*), (Figura 3.5).

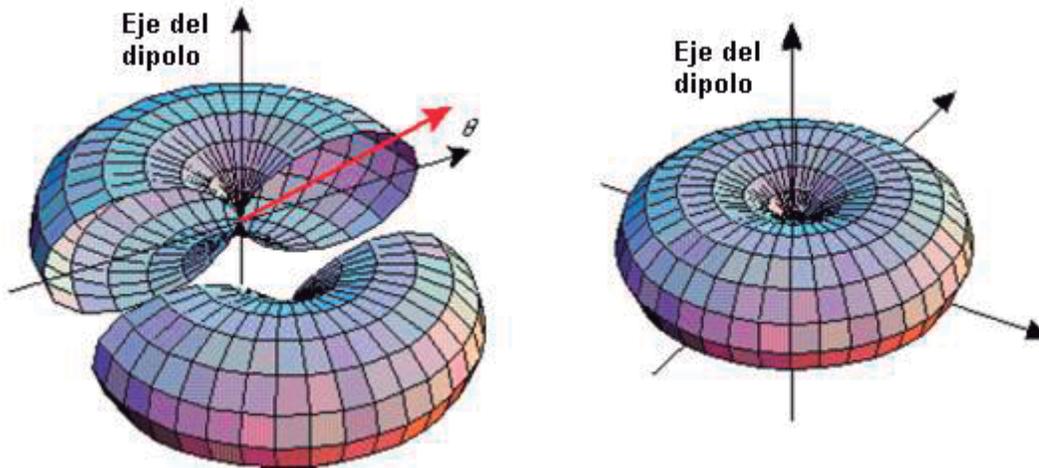


Figura 3.5. Patrón de radiación de antena omnidireccional

Lo más habitual es que los dispositivos inalámbricos tengan instaladas por defecto antenas omnidireccionales. Como no se consigue con éstas un patrón esférico, pero sí se puede controlar el plano donde se coloca el *donut*, el plano se pone paralelo a la horizontal, con lo que la aproximación anterior es válida si la terminal móvil se sitúa en la misma planta que el nodo baliza o en sus proximidades.

La ventaja de este método es que es de aplicación directa en cualquier entorno, suponiendo que el modelo empleado en cada caso se ajuste a la propagación real. Sí la propagación real difiere mucho de la teórica, se tendrá que obtener un modelo adaptado del entorno por lo que la generalidad, ventaja principal del método, se pierde.

Por último, el inconveniente de este enfoque es que al tratar RSSI como distancia, no tienen en cuenta las paredes, mobiliario, suelos, etc. por lo que no se consideran apropiados para el entorno de este caso en particular.

3.2.2 Reconocimiento de patrones

Las técnicas de reconocimiento de patrones, o *fingerprinting*, intentan encajar el vector de potencia recibido (RSSI) procedente de varios puntos, con una muestra de calibración sin necesidad de usar algoritmos geométricos. Para su uso, primero hay que obtener un radio mapa patrón, que en este caso es una base de datos con:

- Posiciones en el edificio
- Niveles RSSI de cada punto de acceso en esas posiciones.

El valor RSSI se puede calcular de forma teórica, con el uso de un modelo de predicción de potencias, o bien se puede obtener de forma práctica midiendo en cada punto de interés la potencia recibida.

La ventaja de este método frente a la triangulación es que los cálculos son más reducidos, y a diferencia del anterior, la discretización del entorno hace que no sea posible obtener una localización que no se haya definido; en cambio, en la triangulación se puede localizar en cualquier punto del espacio por no haberse discretizado la planta.

El grado de discretización de la planta es importante, ya que cuanto más se divida el espacio de la planta, mayor es el grado de precisión, para conseguirlo, se necesita mayor tiempo de procesamiento y de almacenamiento, pero de no contar con un algoritmo de localización suficientemente robusto también será mayor el error cometido.

Los cálculos que permiten la estimación de la localización en los algoritmos de reconocimiento de patrones son sencillos y cambian según sea el método a emplear, pero se basan en, dado un vector de potencias, encontrar la localización que tenga el vector de potencias más parecido.

4. Descripción del protocolo estándar IEEE 802.15.4

Se sabe que un protocolo de comunicaciones define cuándo y cómo los dispositivos se comunican. El IEEE 802.15.4 es un protocolo de paquetes de datos para redes inalámbricas ligeras, define las capas de nivel físico y Control de Acceso al Medio en un modelo de red.

4.1 Características y especificaciones

El grupo de trabajo IEEE 802 está especializado en redes inalámbricas de área personal (Wireless Personal Area Networks, WPAN). El cual se divide a su vez en subgrupos, Algunos estos son:

- 802.11 se encarga de las WLAN
- 802.15 para las WPAN.

Como en el presente trabajo se tiene un interés en redes WPAN se consideran entre los más importantes grupos de trabajo de 802.15 a los siguientes:

- 802.15.1 que se centra en WPAN/Bluetooth
- 802.15.3 WPAN que se centra en proporcionar alta tasa de transferencia de datos
- 802.15.4 que se enfoca principalmente en proporcionar las siguientes características:

- Bajo consumo de energía: Esto se logra porque los dispositivos se encuentran en modo dormido el 99.9% del tiempo.
- Baterías con larga duración: Esto se logra con poca energía de transmisión y muy pocos ciclos de servicio.
- Baja tasa de transferencia de datos.
- Sistemas de bajo costo.

El estándar no define niveles superiores ni subcapas de interoperabilidad. Existen extensiones, como la especificación ZigBee, que complementan al estándar en la propuesta de soluciones completas.

IEEE 802.15.4 es la base sobre la que se define la especificación de ZigBee, cuyo propósito es ofrecer una solución completa para este tipo de redes, construyendo los niveles superiores de la pila de protocolos que el estándar no cubre.

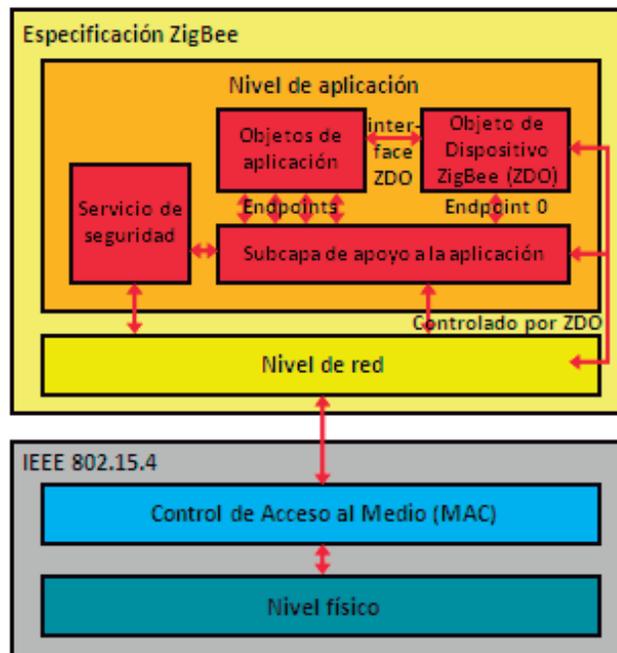


Figura 4.1. Protocolo IEEE 802.15.4

El propósito del estándar es definir los niveles de red básicos para dar servicio a un tipo específico de red inalámbrica de área personal (WPAN) centrada en la habilitación de comunicación entre dispositivos con bajo costo y velocidad.

Entre los aspectos más importantes se encuentra la adecuación de su uso para tiempo real por medio de ranuras de tiempo garantizadas, evitando colisiones por CSMA/CA y soporte integrado a las comunicaciones seguras. También se incluyen funciones de control de consumo de energía como calidad del enlace y detección de energía.

4.1.1 Capa Física

El nivel físico (PHY) proporciona el servicio de transmisión de datos sobre el medio físico propiamente dicho, así como la interfaz con la entidad de gestión del nivel físico, por medio de la cual se puede acceder a todos los servicios de gestión del nivel y que mantiene una base de datos con información de redes de área personal relacionadas. De esta forma, PHY controla el transceptor de radiofrecuencia y realiza la selección de canales junto con el control de consumo de la señal.

Este estándar utiliza la banda ISM (Industrial, Scientific & Medical) para usos industriales, científicos y médicos; en concreto, 868 [MHz] en Europa, 915 [MHz] en Estados Unidos y 2,4 [GHz] en todo el mundo.

La tasa de transferencia de datos depende de la banda de frecuencias a la que se opere. Si se trabaja en la banda ISM de 2.4 [GHz] la tasa de transmisión es de 250 [Kbps], si la banda es de 915 [MHz], la tasa de transmisión es de 40 [Kbps] y si la banda es de 868 [MHz], la tasa de transmisión es 20 [Kbps].

Tabla 4-1 Bandas de Frecuencia para el estándar IEEE 802.15.4.

Bandas de frecuencias (MHz)	Frecuencias [MHz]	Cobertura	Modulación	Transferencia de datos [kbps]	Número de canales
868 / 915	868-868.6	Europa	BPSK	20	1
	902-928	USA	BPSK	40	10
2450	2400-2483.5	Resto del mundo	O-QPSK	250	16

El número de canales de transmisión también depende de la banda utilizada, existen 16 canales para 2.4 [GHz] entre los 2.4 y los 2.4835 [GHz], 10 canales para 915 [MHz] entre los 902.0 y 928.0 [MHz] y un canal si la comunicación se realiza en la banda de 868 [MHz] entre los 868 y los 868.6 [MHz]. En la Figura 4.2 se muestra el número de canales de comunicación para las distintas frecuencias de operación.

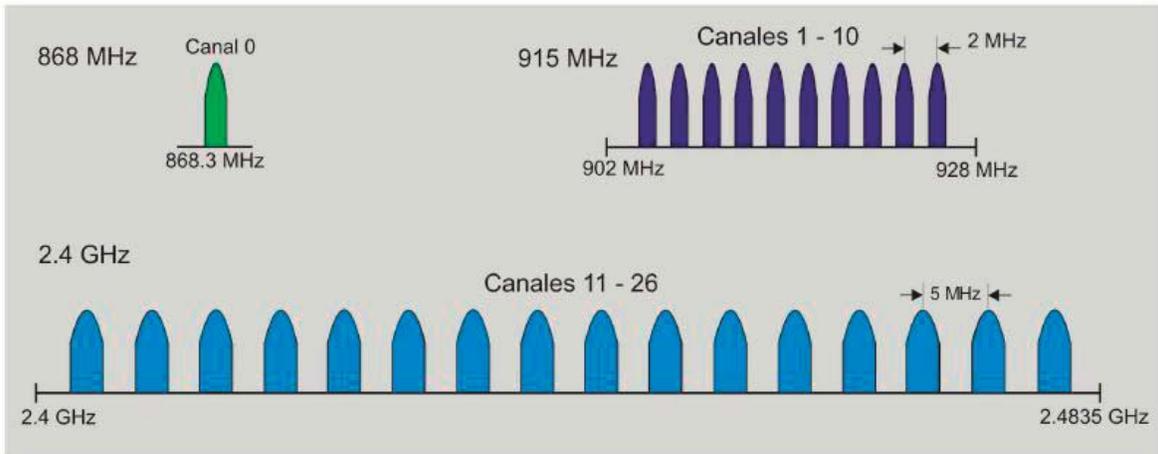


Figura 4.2. Canales de Comunicación IEEE 802.15.4.

Tabla 4-2. Frecuencia de los canales IEEE 802.15.4.

Número de Canales	Frecuencia central del Canal [MHz]
$k = 0$	868.3
$k = 1, 2, \dots, 10$	$906 + 2(k - 1)$
$K = 11, 12, \dots, 26$	$2405 + 5(k - 11)$

La banda de 2.4 [GHz] es la más popular a nivel internacional y la más utilizada en los últimos años por poseer las siguientes características: Espacio entre Canales de 5 [MHz], ancho de banda de cada canal de 3 [MHz], tasa de transferencia de 250 [Kbps], sensibilidad en el receptor que puede alcanzar los -92 [dBm] y un rango típico de alcance en interiores de 20 a 30 [m].

Por otro lado las bandas de 868 [MHz] y 915 [MHz] ofrecen una alternativa a la cogestión creciente y demás interferencias (hornos de microondas, entre otros.) asociadas a la banda de 2.4 [GHz]. Y mayores rangos por enlace debido a que existen menores pérdidas de propagación.

Se utiliza la modulación OQPSK (Offset Quadrature Phase Shift Keying), consiste en realizar una transición de fase en cada intervalo de señalización de bits, por portadora en cuadratura. Aunque también presenta modulación BPSK (Binary Phase Shift Keying) de acuerdo a la frecuencia que se esté usando, se tiene como resultados posibles dos fases de salida para la portadora con una sola frecuencia. Una fase de salida representa un 1 lógico y la otra un 0 lógico. Conforme la señal digital de entrada cambia de estado, la fase de la portadora de salida se desplaza entre dos ángulos que están 180° fuera de fase.

4.1.2 Capa de enlace

El grupo IEEE 802 divide a la capa de enlace en dos sub capas: la de control de enlace lógico (Logical Link Control, LLC) y la subcapa de control de acceso al medio (Medium Access Control, MAC).

En este nivel encontramos el Control de Acceso al Medio (MAC). Éste transmite tramas MAC usando para ello el canal físico. Además del servicio de datos, ofrece un interfaz de control y regula el acceso al canal físico y al balizado de la red. También controla la validación de las tramas y las asociaciones entre nodos, y garantiza slots de tiempo.

En el nivel de enlace de datos podemos localizar el Control de Enlace Lógico (LLC) que hace la función de interfaz con los niveles superiores de la pila de protocolos.

En la Figura 4.3 podemos observar los niveles definidos por el estándar 802.15.4 y las capas intermedias que posibilitan la comunicación con los niveles superiores.

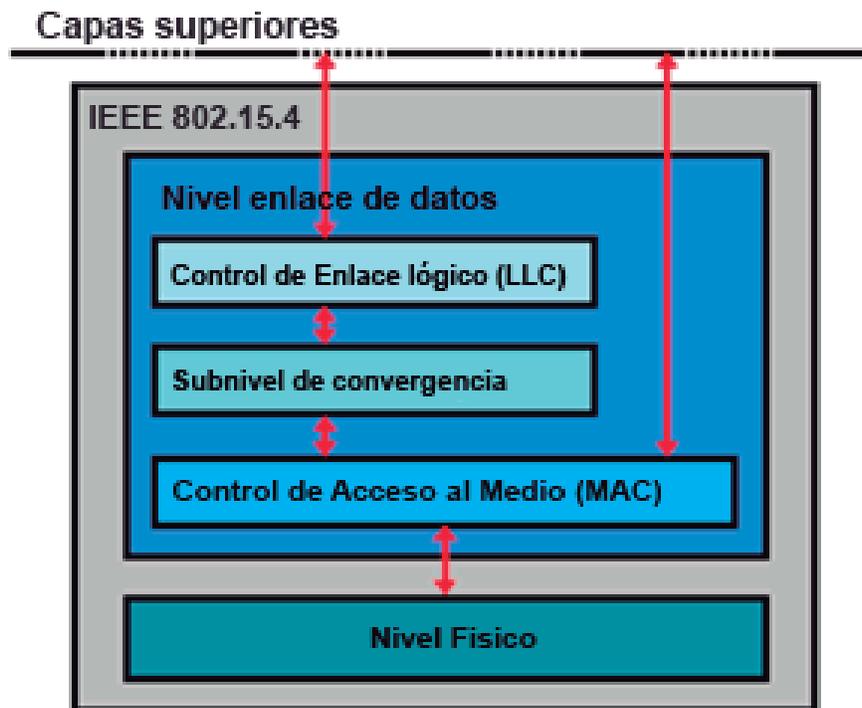


Figura 4.3. Pila de protocolos IEEE 802.15.4.

Las características del MAC IEEE 802.15.4 son: la asociación, la disociación, reconocimientos de entrega de trama, mecanismos de acceso al canal, validación de trama y garantía del manejo de las ranuras de tiempo.

4.2 ZigBee

ZigBee constituye una de las grandes partes del proyecto puesto que es la tecnología de transmisión elegida para el envío de los datos. ZigBee puede ser usado para una gran variedad de aplicaciones, como lo muestra la Figura 4.4, pero en este trabajo interesa su utilización para fines de monitoreo.

El principal objetivo que pretende satisfacer una red de comunicación ZigBee es la de comunicar aplicaciones que requieren comunicaciones seguras, con baja tasa de envío y maximización de la vida útil de sus baterías.

La red, en su conjunto, utilizará una cantidad muy pequeña de energía, de forma que cada dispositivo individual pueda tener una autonomía de hasta 5 años antes de necesitar un recambio en su sistema de alimentación.



Figura 4.4. Aplicaciones ZigBee.

ZigBee está basado en el estándar IEEE 802.15.4 para redes de área personal (PAN) diseñado por la ZigBee Alliance. Se puede definir como una pila o conjunto de protocolos, que permiten la comunicación de forma sencilla entre múltiples dispositivos.

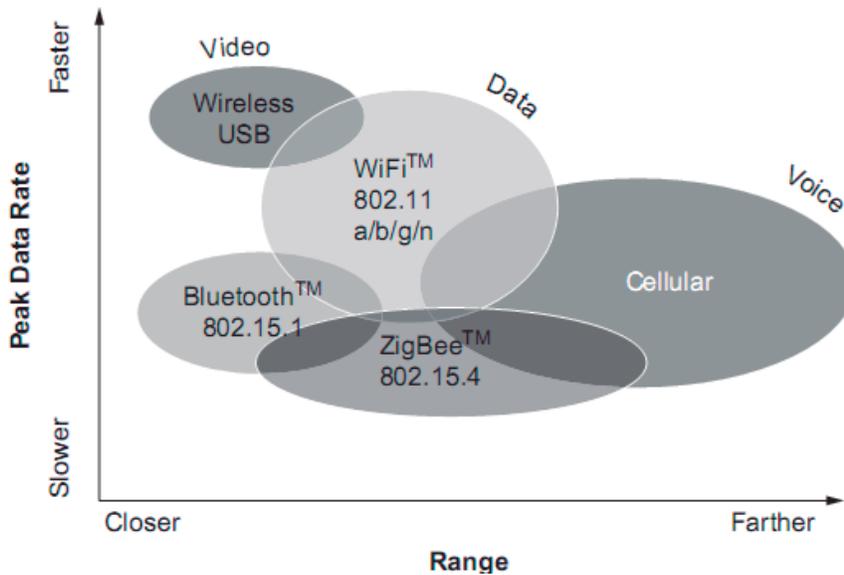


Figura 4.5. Comparación de tecnologías Inalámbricas.

Como puede observarse en la Figura 4.5, ZigBee está por debajo en cuanto a tasas de transmisión, pero también en cuanto a consumo, factor que es muy atractivo. Aunado a estas descripciones se encuentra la distancia a la que puede comunicarse, superando incluso a la tecnología WiFi que es ampliamente utilizada.

El protocolo ZigBee también conocido como ZigBee Stack, utiliza como base el estándar IEEE 802.15.4 el cual define las capas de control de acceso al medio (MAC) y física (PHY) para redes inalámbricas de área personal de baja tasa de transferencia (LR-WPAN). Por lo que ZigBee se enfocó en construir, por encima, la especificación de la capa de red, la especificación de la capa de aplicación y la especificación de servicios de seguridad.

Además de añadir dos capas de alto nivel (nivel de red y de aplicación) a la pila de protocolos, el principal cambio es la adición de los ZDO (ZigBee Device Objects) ya que son los responsables de llevar a cabo una serie de cometidos, entre los que se encuentran: el mantenimiento de los roles de los dispositivos, la gestión de peticiones de unión a una red, el descubrimiento de otros dispositivos y la seguridad. De igual manera hace referencia a los objetos de aplicación definidos

por el fabricante, los cuales permiten la personalización y adaptación a las necesidades del mercado.

La especificación ZigBee completa define cuatro componentes principales:

- Nivel de red.
- Nivel de aplicación.
- Objetos de dispositivo ZigBee (ZDO, ZigBee Device Objects).
- Objetos de aplicación definidos por el fabricante.

En la Figura 4.6 se observa de forma esquematizada, los distintos niveles del estándar 802.15.4 y de la especificación ZigBee.

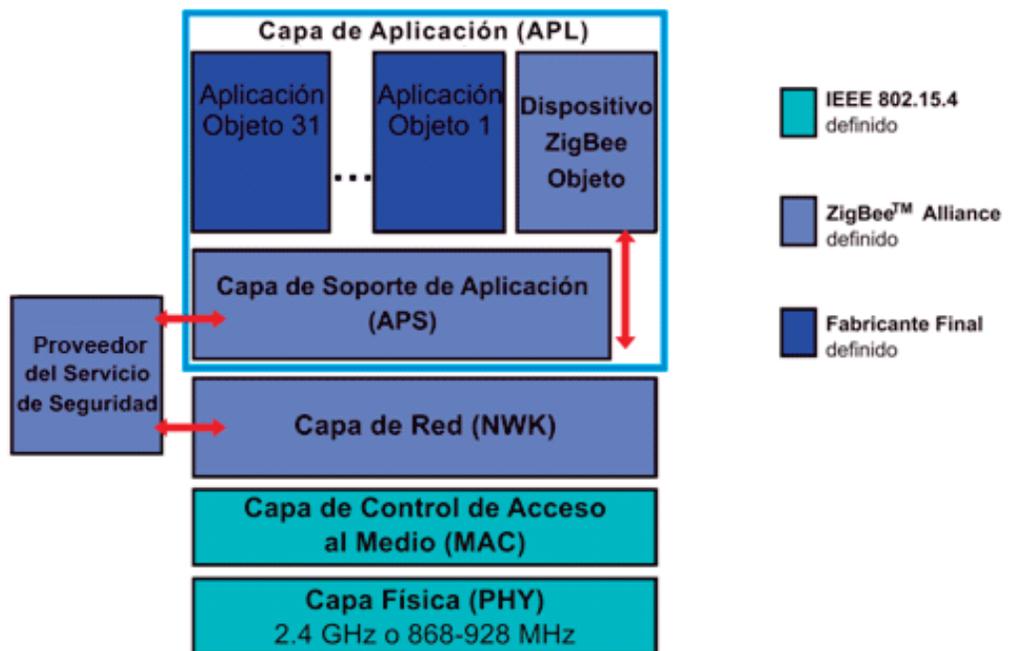


Figura 4.6. ZigBee Stack.

4.2.1 Nivel de red

Las funciones principales del nivel de red son dar paso al correcto uso del Subnivel MAC, definido anteriormente por el estándar IEEE 802.15.4, así como ofrecer una interfaz ajustada para su uso por parte del nivel superior, el nivel de aplicación. Sus capacidades son las típicas de un nivel de red clásico.

Por una parte, la entidad de datos crea y gestiona las unidades de datos del nivel de red a partir del *payload* del nivel de aplicación y realiza el ruteo en base a la topología de la red en la que el dispositivo se encuentra. Por otra parte, las funciones de control del nivel de red controlan la configuración de nuevos dispositivos y el establecimiento de nuevas redes; puede decidir si un dispositivo colindante pertenece a la red e identifica nuevos *routers* y vecinos. El control puede detectar así mismo la presencia de receptores, lo que posibilita la comunicación directa y la sincronización a nivel MAC.

4.2.2 Nivel de aplicación

Se trata del nivel más elevado definido por la especificación y, por tanto, la interfaz efectiva entre el nodo ZigBee y sus usuarios. En él se ubican la mayor parte de los componentes definidos por la especificación: tanto ZDO's, (ZigBee Device Objects) quienes se encargan de definir el papel del dispositivo en la red, si el actuará como coordinador, *router* o dispositivo final como sus procedimientos de control y los objetos de aplicación.

Un nodo está formado por un transceptor compatible con 802.15.4 y uno o más tipos de dispositivos conformados por un conjunto de atributos que pueden ya sea monitorearse o controlar por medio de eventos.

4.2.3 Dispositivos ZigBee

Dentro de una red constituida por esta tecnología se pueden encontrar tres tipos de dispositivos diferentes que cumplen con funciones diferentes de acuerdo a la posición en que se encuentran dentro de una red, dichos dispositivos son los que a continuación se mencionan:

- **Coordinador ZigBee (*ZigBee Coordinator*, ZC):** Consiste en el dispositivo más completo de los tres, puesto que sus funciones son las de controlar y coordinar la red y los caminos que deben seguir los dispositivos para conectarse entre ellos. Entre sus funciones se encuentra establecer el canal de comunicaciones y el PAN ID (Identificador de red) para toda la red, se encarga de formar la red, permitiendo unirse a él a dispositivos Router ZigBee y ZigBee End Device. Debemos encontrar obligatoriamente un ZC en cada red ZigBee.
- **Router ZigBee (*ZigBee Router*, ZR):** crea y mantiene información sobre la red para determinar la mejor ruta para transmitir un paquete de información, además es el encargado de la interconexión de los dispositivos separados en la topología de la red.
- **Dispositivo Final (*ZigBee End Device*, ZED):** En este dispositivo quedan representadas las principales características de ZigBee, como son el bajo consumo de energía y el bajo costo. Los ZED poseen la funcionalidad necesaria para comunicarse con su nodo padre, que ya puede ser el Router ZigBee o el Coordinador ZigBee, pero no puede transmitir información destinada a otros dispositivos. Es por ello, que este tipo de dispositivo puede estar en modo dormido la mayor parte del tiempo, aumentando así la vida media de sus baterías.

El estándar 802.15.4 define dos dispositivos según las funciones que estos pueden realizar en una red ZigBee:

- **Dispositivo de Funcionalidad Completa (FFD, *Full-Function Device*):** Conocido también como nodo activo. Gracias a la memoria adicional y a la capacidad de cómputo puede funcionar como Coordinador o Router ZigBee

de una red de área personal (PAN) o como un nodo normal. Puede ser usado en dispositivos de red que actúen de interface con los usuarios.

- Dispositivo de Funcionalidad Reducida (RFD, *Reduced-Function Device*): Conocido también como nodo pasivo. Posee una capacidad y funcionalidad limitada para garantizar un bajo costo y una gran simplicidad, por ello sólo pueden comunicarse con FFD's y nunca pueden ser coordinadores. Básicamente constituyen los sensores de la red.

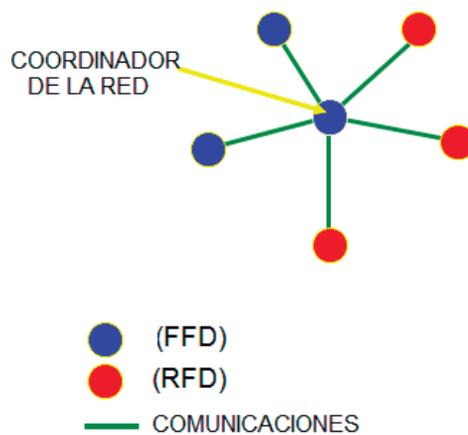


Figura 4.7. Dispositivos FFD y RFD.

La forma de configurar o programar ZigBee, depende del fabricante, puede ser en lenguaje C, o simplemente comandos AT o de otra manera, también dependerá de si se está utilizando un módulo con todas las características integradas (MCU + radio) o solo el dispositivo de radiofrecuencia (radio), etc.

4.2.4 Seguridad

A pesar de no ser una capa como tal, esta funcionalidad coexiste con el estándar en busca de factores esenciales a tomar en cuenta en la comunicación; la confidencialidad de la información para lo cual maneja AES⁴ como algoritmo de encriptación, y para integridad de la información añade a cada trama saliente un código de integridad de mensaje (MIC⁵). El código MIC permite distinguir si la información ha sido alterada y se usa para cuidar modificaciones de los mensajes o reenvío de mensajes anteriores por parte de un intruso.

⁴ Estándar de cifrado de bloque, con bloques de 128 bits y llaves de 128, 192 ó 256 bits.

⁵ Secuencia de bits que ayudan a detectar si un mensaje ha sido modificado

5. Diseño del hardware

ZigBee es un estándar de comunicaciones creado por la Alliance ZigBee, la cual está constituida por más de cien compañías y cada una ofrece en el mercado su producto a base de la pila de protocolo ZigBee (estándar IEEE 802.15.4).

El módulo transceptor que se seleccionó es el XBee IEEE® 802.15.04 de la casa MaxStream compañía que forma actualmente parte de DIGI International.

5.1 Módulos XBee

La compañía MaxStream proporciona un transceptor de nombre XBee IEEE® 802.15.04 (Figura 5.1), que opera en la banda de 2.4 [GHz] para la integración de los sistemas que buscan transmitir información por medio de RF de manera segura con una baja tasa de transferencia de información, así como tratando de maximizar la vida útil de las baterías.

Los módulos XBee pueden ser configurados externamente y llevar a cabo acciones de comandos AT y específicos de control. Debido a esto están diseñados para interactuar con microcontroladores, a través de una interfaz SPI⁶, con un máximo de recepción y transmisión de datos.

⁶ SPI (Serial Peripheral Interface Bus o bus serial de interfaz de periféricos) es un estándar de enlace de datos seriales sincronizados por un reloj que operan en modo full duplex.

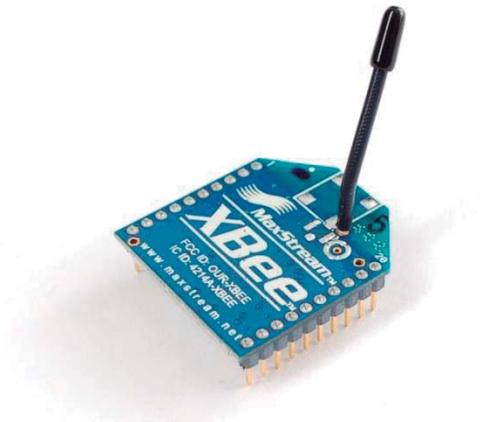


Figura 5.1. Módulo XBee.

Los módulos XBee físicamente son pequeños y se encuentran en dos versiones diferentes (XBee / XBee-Pro), dependiendo del poder de transmisión. En la Figura 5.2 se puede apreciar las dimensiones físicas para ambas versiones.

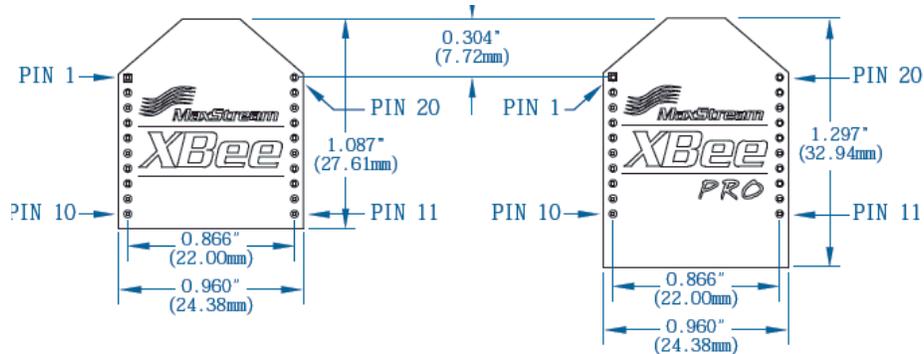


Figura 5.2. XBee/XBee-Pro. (Tomada del Datasheet de los módulos XBee).

En la Tabla 5-1 se aprecian las diferencias técnicas existentes entre las dos versiones disponibles en el mercado para los dispositivos XBee y XBee-Pro.

Tabla 5-1. Especificaciones para XBee/XBee-PRO OEM RF Modules.

Especificaciones	XBee	XBee-PRO
Desempeño		
Indoor / Rango Urbano	Hasta 100ft (30m)	Hasta 300ft (100m)
Outdoor RF línea de Vista	Hasta 300ft (100m)	Hasta 1 milla (1500m)
Potencia de transmisión	1mW (0 dBm)	60mW (18 dBm) guiado, 100mW (20 dBm) EIRP.
Tasa de transferencia de Datos RF	250,000 bps	250,000 bps
Tasa de transferencia de datos con la Interfaz (elegible por Software)	1,200–115,200 bps (también soporta velocidades de baudios no estándares)	1,200–115,200 bps (también soporta velocidades de baudios no estándares)
Sensibilidad del receptor	-92 dBm (1% de promedio de error)	-100 dBm (1% de promedio de error)
Requerimientos de Energía		
Voltaje de alimentación	2.8 ~ 3.4V	2.8 ~ 3.4V
Corriente (típica) del transmisor	45mA (@3.3V)	270mA (@3.3V)
Corriente (típica) del receptor	50mA (@3.3V)	55mA (@3.3V)
Corriente de apagado	< 10µA	< 10µA
General		
Frecuencia de operación	ISM 2.4 GHz	ISM 2.4 GHz
Dimensiones	0.960" x 1.087" (2.438cm x 2.761cm)	0.960" x 1.297" (2.438cm x 3.294cm)
Temperatura de operación	-40° a 85°C (industrial)	-40° a 85°C (industrial)
Opciones de antena	Conector U. FL, antena de circuito y antena Whip.	Conector U. FL, antena de circuito y antena Whip.
Red y seguridad		
Topologías de red soportadas	Punto a punto, punto a multipunto, Peer-to-peer.	Punto a punto, punto a multipunto, Peer-to-peer.
Número de canales (elegibles por software)	16 canales de secuencia directa	13 canales de secuencia directa
Opciones de filtrado	PAN ID: Canal y fuente/dirección destino	PAN ID: Canal y fuente/dirección destino.

Existen 2 series de estos módulos: La serie 1 y la serie 2 o también conocida como 2.5. En el mercado actualmente se encuentra la serie 2 que ha mejorado sus características técnicas, pero su costo es alto.

Los módulos de la Serie 1 y la Serie 2 tienen el mismo pin-out (Tabla 5-2), sin embargo, no son compatibles entre sí ya que utilizan distintos chipset y trabajan con protocolos diferentes. La serie 1 está basada en el chipset Freescale y está pensado para ser utilizado en redes punto a punto y punto a multipunto mientras que los módulos de la serie 2 están basados en el chipset de Ember y están

diseñados para ser utilizados en aplicaciones que requieren repetidores o una red *mesh*.

Los módulos XBee han sido diseñados para ser montados directamente sobre un soporte. Por esta razón, no requieren ningún tipo de soldadura en la instalación. Estos además disponen de múltiples pines para la conexión, cumpliendo diversas funciones: alimentación, conexión a tierra, entrada de datos, etc.

A continuación la Tabla 5-2 muestra la funcionalidad de cada uno de los 20 pines que contiene el módulo XBee:

Tabla 5-2. Descripciones de los pines para XBee and XBee-PRO Modules.

Pin #	Nombre	Dirección	Descripción
1	VCC	-	Fuente de alimentación
2	DOUT	Salida	Datos de salida UART
3	DIN/CONFIG	Entrada	Datos de entrada en UART
4	DO8 *	Salida	Salida Digital 8
5	RESET	Entrada	Módulo de entrada de reinicio (reset pulso debe ser de al menos 200 ns)
6	PWM0/RSSI	Salida	Salida PWM 0/indicador RX de intensidad de la señal
7	PWM1	Salida	Salida PWM 1
8	[reservado]	-	No conecte
9	DTR/SLEEP_RQ/DI8	Entrada	pin de de control sleep o Entrada Digital 8
10	GND	-	Tierra
11	AD4/DIO4	Ambos	Entrada analógica 4 o Digital I/O 4
12	CTS/DIO7	Ambos	clear-to-send Flow control o Digital I/O 7
13	ON/SLEEP	Salida	Indicador del estado del módulo
14	VREF	Entrada	Voltaje de referencia para entradas A/D
15	Asociado/AD5/DIO5	Ambos	Indicador de asociación, entrada analógica 5 o Digital I/O 5
16	RTS/AD6/DIO6	Ambos	Request-to-send flow control, entrada analógica 6 o Digital I/O 6
17	AD3/DIO3	Ambos	Entrada analógica 3 o Digital I/O 3
18	AD2/DIO2	Ambos	Entrada analógica 2 o Digital I/O 2
19	AD1/DIO1	Ambos	Entrada analógica 1 o Digital I/O 1
20	AD0/DIO0	Ambos	Entrada analógica 0 o Digital I/O 0

* La función no se admite en el momento de esta versión

XBee, es una tecnología inalámbrica con tasas de transmisión comprendidas entre los 20 [Kbps] y 250 [Kbps]. En cuanto a la potencia de los módulos se asocia un valor de 1 [mW] lo que permite alcanzar una distancia entre 30 [m] y 100 [m] (exterior / interior). En cuanto al consumo, estos dispositivos tienen un consumo muy bajo (inferior de 10 [μ A]) cuando se encuentra en estado de reposo.

Dichos módulos se encuentran en tres opciones diferentes según el tipo de antena: Whip antenna, Chip antenna y U.FL.RF connector. En el proyecto se utilizarán los módulos con antena Whip debido a que para este tipo de red (bajo techo) mejora el rendimiento de la red.

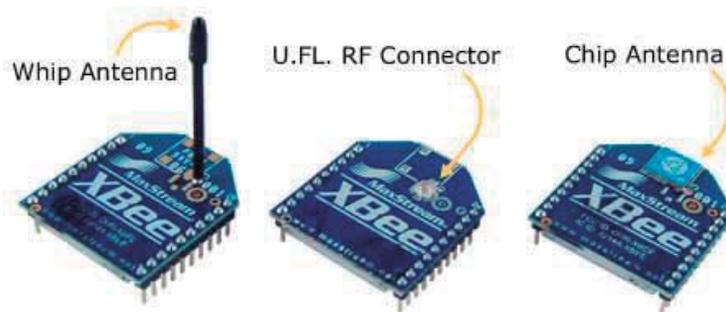


Figura 5.3. Tipos de antenas.

Los módulos XBee están implementados sobre un ZigBee SoC (System on Chip) EM250 de Ember el cual incluye un transceptor de 2.4GHz y un microprocesador XAP2b de 16 bits. Los módulos ofrecen pines destinados a GPIO (Entrada Salida de Propósito General) que pueden configurarse como entradas/salidas digitales con o sin pull-up, ADCs o PWMs en uno de los cuales se puede obtener la intensidad de la señal recibida. Por defecto los pines vienen como Output/Input, para funcionar como entradas analógicas. Además para facilitar el desarrollo el fabricante provee un Stack ZigBee (EmberZNet PRO) sobre el que se implementó dos modos de control: comandos AT y tramas API.

El módulo podrá comportarse como Coordinador, Router o End Device según el firmware⁷ instalado. Digi provee un software para Windows llamado XCTU que permite grabar el firmware y configurar los parámetros, el cual fue utilizado para la implementación.

Son muchas las aplicaciones en las que se han podido implementar estos módulos, gracias a las especificaciones y características técnicas, las cuales presentan una calidad superior sobre sus competidores que de igual manera trabajan sobre el estándar ZigBee.

5.1.1 Modos de operación

El dispositivo cuenta con los siguientes modos de operación:

- Idle Mode: cuando no se está recibiendo o transmitiendo información.
- Transmit Mode: al recibir información serial en el DI Buffer.
- Receive Mode: recepción de un paquete valido de RF a través de la antena.
- Sleep Mode: estado de bajo consumo de energía.
- Command Mode: se ejecuta una secuencia de caracteres que son interpretados como un comando.

5.1.2 Comunicación Serial

El módulo XBee posee una interfaz física de comunicaciones serial asincrónica para conectarse a un dispositivo. A través de su puerto serial, el módulo puede comunicarse con cualquier lógica y voltaje compatible con la UART.

⁷ Bloque de instrucciones que definen la lógica a bajo nivel para controlar un circuito electrónico

Los dispositivos que disponen de una interfaz UART se pueden conectar directamente a los pines del módulo de RF como se muestra en la siguiente Figura 5.4.

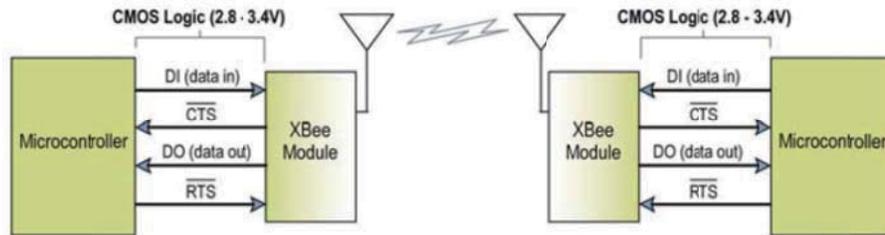


Figura 5.4. Comunicación UART. (Tomada del Datasheet de los módulos XBee).

Los datos ingresan al módulo UART a través del pin DI-Data in (pin 3) como una señal serial asíncrona. La señal debe estar en alto cuando no se están transmitiendo datos. Cada byte de datos se compone de un bit de inicio (Low), 8 bits de datos (bit menos significativo primero) y un bit de parada (High). La siguiente Figura 5.5 ilustra el patrón del bit serial de datos que pasan a través del módulo.

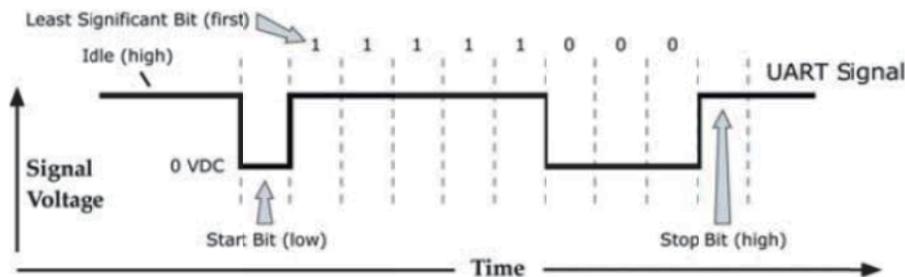


Figura 5.5. Paquete de datos UART, transmitido a través del módulo de RF. (Tomada del Datasheet del módulo XBee).

El módulo UART realiza las tareas, tales como sincronización y la comprobación de paridad, que se necesitan para la comunicación de datos. La comunicación serial dependerá de que los dos UARTs (XBee y periférico) tengan la configuración compatible (velocidad, paridad, bits de inicio, bits de parada, bits de datos).

5.1.3 Operación transparente

Todos los datos recibidos a través de un pin 3 del chip llamado “Data Input” son colocados en el buffer de entrada para su posterior retransmisión por radiofrecuencia. Cuando un módulo recibe un paquete vía radiofrecuencia, los datos son enviados por el pin 2 “Data Output” sin añadirle ningún tipo de información adicional.

Al funcionar en modo transparente, los módulos se configuran mediante los comandos AT por lo que los datos son enviados a un módulo definido previamente en la configuración. Este modo está destinado principalmente a la comunicación punto a punto, donde no es necesario ningún tipo de control. También se usa para reemplazar alguna conexión serial por cable.

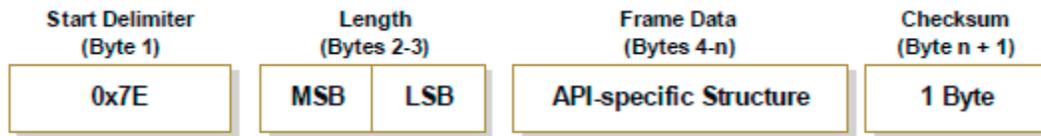
5.1.4 Operación API

Este modo es más complejo, pero permite el uso de *frames* con cabeceras que aseguran la transmisión de los datos, al estilo TCP. Cuando el módulo XBee se encuentra en este modo, toda la información que entra y sale, es empaquetada en frames, que definen operaciones y eventos dentro del módulo

El modo de operación API facilita muchas operaciones, tales como:

- Transmisión de datos hacia múltiples destinatarios sin ingresar al modo Command.
- Estatus de recepción exitosa o fallida para cada uno de los paquetes RF transmitidos.
- Identificación de las direcciones fuente de cada paquete recibido.

El modo API se establece con la siguiente estructura:



MSB = Most Significant Byte, LSB = Least Significant Byte

Figura 5.6. Estructura general de una trama en modo de operación API. (Tomada del Datasheet del módulo XBee).

A continuación se detalla la función de los campos que son comunes en todos los tipos de mensajes.

- Delimitador de Inicio, está formado por una secuencia de 8 bits que indica el inicio de la trama, la secuencia es 0x7E. Cualquier dato recibido antes del delimitador de inicio es descartado.
- Longitud, indica el número de octetos contenidos dentro del campo de datos de la trama. Este campo está conformado por dos bytes, donde el primer octeto es el más significativo y el segundo el menos significativo.
- Suma de Verificación (Checksum), este campo está formado por un octeto, permite verificar la integridad de los datos a la unidad receptora. Para calcular el valor que debe ser asignado a este campo, debemos sumar todos los octetos correspondientes al campo de datos de la trama (no se incluyen el delimitador de inicio y la longitud), si el resultado de la suma es superior a 0xFF, solamente debemos tomar los 8 bits menos significativos, el resultado obtenido debemos restar de 0xFF, el resultado de esta operación es el valor que debe ir en el campo de suma de verificación. Para comprobar si el valor del campo de suma de verificación es correcto, se debe sumar todos los octetos de la trama menos el delimitador de inicio y la longitud, si el resultado de esta suma es 0xFF el valor es correcto, caso contrario no lo es.
- En el campo de datos de la trama, se guarda un determinado tipo de mensaje especificado por el identificador API.

En la Tabla 5-3, se indica el valor del identificador API (hexadecimal) y el tipo de mensaje que corresponde.

Tabla 5-3. Identificador API y su tipo de mensaje.

Identificador	Tipo de mensaje
0x8A	Estado del módulo
0x08	Comando
0x09	Aplicar un comando
0x88	Repuesta a un comando
0x00	Requerimiento de transmisión (dirección 64 bits)
0x01	Requerimiento de transmisión (dirección 16 bits)
0x89	Estado de la transmisión
0x80	Paquete recibido (dirección 64 bits)
0x81	Paquete recibido (dirección 16 bits)

El enlace inalámbrico requiere el uso de tres tipos de mensajes en particular: un mensaje para solicitar la transmisión de datos, un mensaje que indica la recepción de un paquete y un mensaje que indica el estado de un paquete transmitido. Por lo que ahora se pasará a descomponer el paquete “Frame Data” para el envío y la recepción (usando direcciones de 16 bits) que tendrá significados distintos según se use de un modo u otro, cabe aclarar que solo se revisaran estos *frames* debido a que son los que se utilizaron en la implementación del proyecto.

5.1.5 Petición de envío (TX Request) para direcciones de 16 bits

Una petición de envío hará que el módulo envíe los datos, que han de ser previamente empaquetados. El formato es el que indica la Figura 5.7:

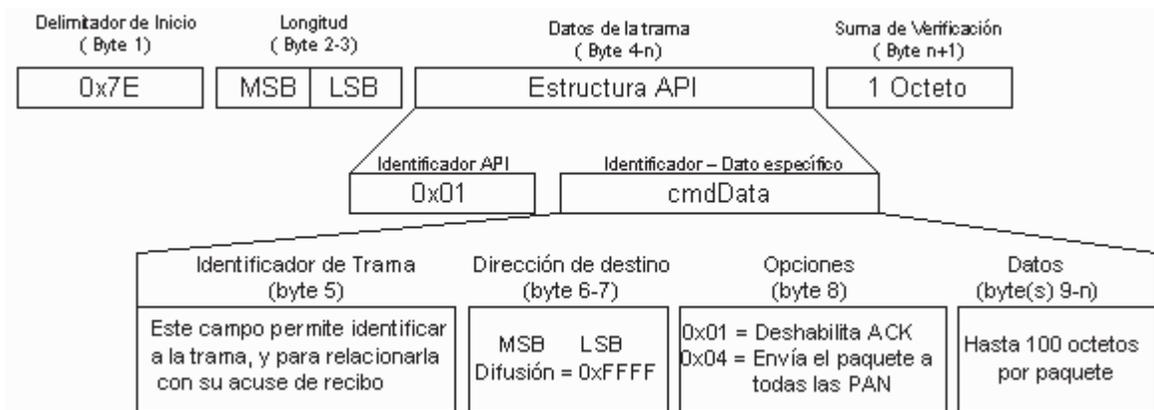


Figura 5.7. Estructura para el envío en modo API usando 16 bits.

Dentro de los Datos de la trama se pueden distinguir los siguientes campos:

- Identificador API, este campo está conformado por un octeto, el cual indica el tipo de mensaje. El identificador para una trama de solicitud de transmisión usando direcciones cortas de 16-bits, correspondiente es el 0x01 (Tabla 5-3).
- Identificador de Trama, como su nombre lo indica, este campo permite reconocer a la trama, además permite relacionarla con su correspondiente acuse de recibo. Si el valor de este campo es 0, se deshabilita la trama que indica el estado de la trama enviada.
- Dirección de destino, este campo está formado por dos octetos, los cuales indican a quien está dirigida la trama. Primero se debe colocar el octeto más significativo de la dirección y a continuación el menos significativo. Para enviar la trama a todas las unidades se debe utilizar la dirección 0xFFFF.
- Opciones, este campo está formado por un octeto, y permite deshabilitar el acuse de recibo ó enviar la trama a todas las PAN.
- Datos, este campo contiene la información que se desea enviar, puede tener hasta 100 octetos.

5.1.6 Recepción de datos (RX Packet) para direcciones de 16 bits

Cuando un paquete es recibido por un módulo, éste lo envía a través de su puerto serial mediante la estructura indicada en la Figura 5.8.

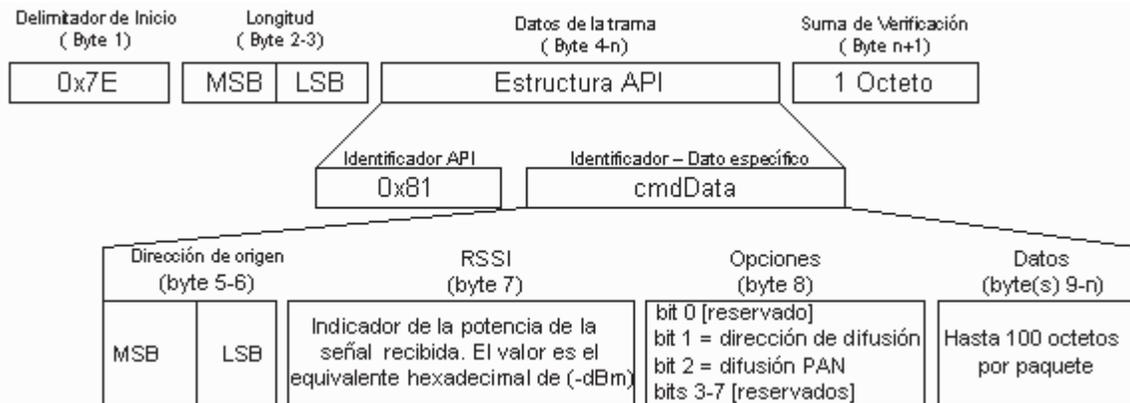


Figura 5.8. Estructura para la recepción en modo API usando 16 bits.

Los campos de la estructura son los siguientes:

- El identificador API correspondiente es el 0x81.
- Dirección de Origen, este campo permite indicar a la unidad receptora quien le envió el paquete, este campo está formado por dos octetos, el primero representa el byte más significativo de la dirección de 16-bits y el segundo el menos significativo.
- El campo RSSI, este campo indica el nivel de potencia con que se recibieron los datos. Por ejemplo, si el valor de este campo es 0x25 (37 decimal), quiere decir que la potencia de la señal recibida es igual a -37dBm.
- Opciones, este campo permite saber a la unidad receptora cómo fue direccionada la trama, este campo está formado por un octeto, el bit 1 indica, si la trama fue enviada con una dirección de difusión (DH=0xFFFF) y el bit 2 indica si la trama fue enviada a todas las PAN (ID PAN=0xFFFF).
- Datos, este campo puede contener hasta 100 octetos, corresponde a los datos recibidos.

5.1.7 Estructura de una trama de estado de paquete.

Este mensaje se genera en respuesta a una solicitud de transmisión. Esta clase de mensaje indica si el paquete enviado fue transmitido exitosamente o si la transmisión falló.

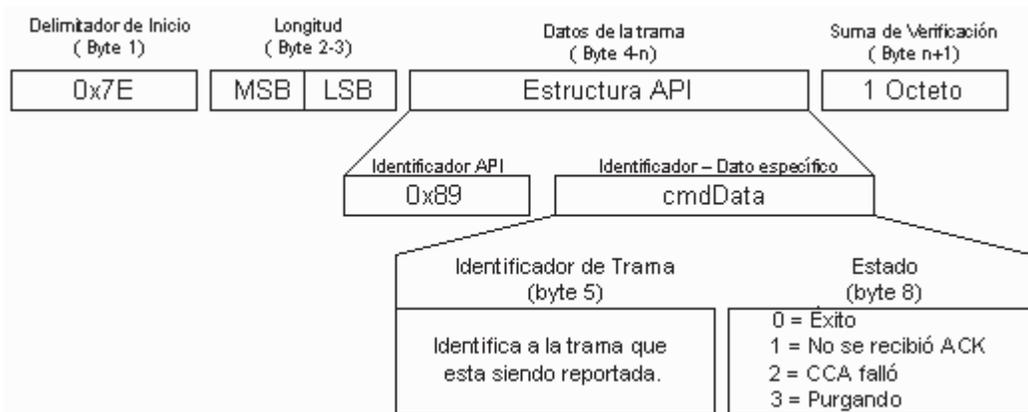


Figura 5.9. Trama que indica el estado del paquete enviado

- El identificador API de trama correspondiente a este tipo de mensaje es 0x89
- Identificador de trama, este campo está formado por un octeto, permite indicar la trama que está siendo reportada. Si el campo de identificador de trama, del paquete enviado es 0x00, no se genera este mensaje.
- Estado, este campo está formado por un octeto, indica el estado de la trama enviada. Si el valor de este campo es 0x00 quiere decir que la transmisión se realizó exitosamente, un valor igual a 0x01 indica que el tiempo de espera para recibir el acuse de recibo expiró (no se recibió el ACK), un valor igual a 0x02 indica que el paquete no se envió porque no se dieron las condiciones adecuadas (el parámetro CA establece que la energía detectada en el canal debe ser superior a -44dBm para enviar un paquete) y por último si el valor es igual a 0x03 quiere decir que el tiempo de espera para recibir una transmisión indirecta expiró.

La operación API es la que nos interesa. Ello es debido a que en el *frame* de la recepción hay dos datos que llegan especialmente importantes: el **identificador del emisor** y el **nivel de señal recibida (RSSI)**. Procesando y analizando estos dos datos que llegan de cada emisor se realizará el posicionamiento en espacios cerrados.

5.1.8 Direccionamiento ZigBee en XBee.

El direccionamiento ZigBee utiliza dos direcciones para cada dispositivo:

- Una de 64 bits (identificador EUI-64) también llamada dirección extendida que es fija y única para cada dispositivo, asignada por el fabricante. Sus primeros 24 bits corresponden a un identificador de vendedor asignado por la IEEE (OUI: Organizationally Unique Identifier). En el caso de los módulos XBee siempre es 00-13-A2 que corresponde a MaxStream Inc. Los siguientes 40 bits son un espacio de direcciones definido por el vendedor que debe encargarse de asignar a cada dispositivo fabricado un valor diferente.
- Una de 16 bits que les es asignada por el coordinador o router al unirse a una red.

Para una transmisión unicast ZigBee necesita ambas direcciones, la de 64 bits para identificar inequívocamente al destinatario y la de 16 bits para rutear el mensaje. Cuando una transmisión sólo especifica la dirección de 64 bits entonces el Stack ZigBee deberá encargarse de descubrir la dirección de 16 bits correspondiente.

5.2 Determinación de módulos a implementar.

Fue necesario analizar y determinar cuál de las dos referencias ofrecidas por la empresa MaxStream de módulos XBee se usaría en la implementación y en el desarrollo de la red inalámbrica para el proyecto. Teniendo en cuenta las características técnicas mostradas en la Tabla 5-1, la diferencia entre las dos versiones se observa en la potencia de salida de transmisión que se ve directamente reflejada en el factor de alcance y la corriente de transmisión/recepción, y de acuerdo con las necesidades del proyecto de cubrir la mayor área posible con el menor número de requerimientos y equipos, se optó por el empleo los módulos XBee-Pro.

Cabe mencionar que se desarrolló el proyecto con la implementación de módulos XBee serie 1 en lugar de la serie 2 debido a que los módulos correspondientes a la serie 2 son más caros respecto a la serie 1, factor de vital importancia ya que esto elevaría el costo total del proyecto, otro factor importante a tener en cuenta es que dado el objetivo del trabajo que es la determinación de la posición de un objeto, las características extras ofrecidas por el XBee serie 2 no se implementarían en su totalidad. Por estas razones, se utilizarán los módulos XBee serie1.

Dispositivo seleccionado:

- XBee Serie 1 OEM RF modelo **XBP24-AWI-001**.

5.3 Interfaz para XBee

Es necesario realizar una tarjeta para cada módulo de la red inalámbrica, teniendo en cuenta que los módulos XBee se van a ubicar en una zona específica deben contar con un microcontrolador que tome los datos recibidos, procese la información y la transmita de nuevo hacia módulo XBee que se desea localizar,

esto con el objetivo de obtener el valor de RSSI respecto a una posición conocida, por lo cual es necesario conectar el radio XBee hacia un dispositivo externo ya que la configuración interna no permite realizar el proceso antes descrito, es entonces aquí donde entra a escena el microcontrolador.

5.4 Desarrollo programa para el Microcontrolador

El módulo XBee de cada zona recibe dos datos: el nivel de sensibilidad de percepción (RSSI) y la dirección del módulo remoto, pero es necesario que esta información sea transmitida hacia la computadora para que ésta determine la posición estimada del objeto.

Para llevar a cabo lo descrito anteriormente es necesario conectar un microcontrolador a cada XBee, ya que el radio por si solo no es capaz de realizar la tarea.

El software a utilizar para programar los microcontroladores es CCS PCWH Compiler. La decisión de optar por este software se debe a que permite programar microcontroladores por medio de un lenguaje casi idéntico al popular C sin necesidad de conocer las arquitecturas de estos.

5.5 Microcontroladores

Un microcontrolador (MCU, microcontroller unit) es una microcomputadora digital de bajo costo integrada en un circuito integrado de silicio que cuentan con: un microprocesador o unidad de procesamiento central (CPU), una memoria para almacenar el programa, una memoria para almacenar datos, generador de reloj, puertos de entrada salida, convertidores A/D (Analógico a Digital) y D/A (Digital a analógico).

Este dispositivo tiene la habilidad de procesar información mediante la ejecución de programas, y mostrar los resultados por los puertos I/O. Pueden conectarse componentes electrónicos, siendo el MCU el responsable de inteligencia para la toma de decisiones. Uno de los microcontroladores más comunes son los PIC, llamados así por Microchip para describir su serie de microcontroladores. Para que un PIC pueda realizar la actividad que queremos que haga, es necesario programarlos. Los lenguajes más comunes son ensamblador y C.

El funcionamiento de los microcontroladores está determinado por el programa almacenado en su memoria. Este puede escribirse en distintos lenguajes de programación. Además, la mayoría de los microcontroladores actuales pueden reprogramarse repetidas veces. Por las características mencionadas y su alta flexibilidad, los microcontroladores son ampliamente utilizados como el cerebro de una gran variedad de sistemas embebidos⁸ que controlan máquinas, componentes de sistemas complejos, como aplicaciones industriales de automatización y robótica, domótica, equipos médicos, sistemas aeroespaciales, e incluso dispositivos de la vida diaria como automóviles, hornos de microondas, teléfonos y televisores.

Básicamente el proceso de desarrollo de una aplicación basada en microcontroladores se compone de las siguientes etapas:

- Desarrollo de software: etapa que corresponde a la escritura y compilación/ensamblaje del programa que regirá las acciones del MCU y los sistemas periféricos conectados a éste.
- Programación del MCU: el código de máquina correspondiente al programa desarrollado en la etapa anterior se descarga en la memoria del MCU.

⁸ Sistema de computación diseñado para realizar una o algunas pocas funciones dedicadas frecuentemente en un sistema de computación en tiempo real.

- Pruebas y verificación: Por último, el MCU debe conectarse al circuito electrónico para el cual fue diseñado y someterse a pruebas para verificar el funcionamiento correcto del programa.

En la etapa del desarrollo de software tenemos distintas maneras de desarrollar el programa dependiendo del lenguaje que se utilizará para escribir el programa. El método básico es escribir el programa en lenguaje Assembly⁹ (lenguaje de Ensamblador) en un archivo de texto con extensión .asm para luego utilizar una programa ensamblador (Assembler) para generar un archivo en lenguaje de máquina, compuesto por instrucciones en código binario que son directamente entendidas por el CPU del microcontrolador.

El lenguaje Assembly se compone de instrucciones de bajo nivel, es decir que están ligadas directamente a las características del microcontrolador por lo cual el lenguaje Assembly es más difícil de emplear, requiere experiencia y un mayor tiempo de desarrollo. La ventaja es que el código generado a partir de un programa escrito en lenguaje de máquina es por lo general más eficiente, debido a que el programa se desarrolla en un nivel cercano a las características del hardware.

La otra alternativa, la cual fue empleada para el desarrollo del proyecto, es emplear un lenguaje de alto nivel, el cual es más fácil de usar y reduce los tiempos de desarrollo debido a que son lenguajes muy parecidos a C y C++ entre los más populares pero también existen otros lenguajes variantes del BASIC y el Pascal.

Por lo que una vez escrito el programa en el lenguaje de alto nivel, será necesario emplear un compilador para traducirlo, ya sea a lenguaje de Ensamblador o directamente a lenguaje de máquina.

⁹ En estricto rigor, Assembly es el lenguaje y Assembler es la herramienta de software que traduce el código Assembly a lenguaje de máquina. Sin embargo, es normal en el uso profesional emplear la palabra Assembler en forma ambigua para referirse tanto al lenguaje de programación como a la herramienta de software ensambladora.

Es importante considerar que el código de Ensamblador generado por los compiladores tiende a ser más largo e ineficiente que aquel directamente desarrollado en lenguaje de Ensamblador, esta desventaja puede ser crítica en ciertas aplicaciones que requieren programas compactos y de una alta velocidad de ejecución. La Figura 5.10. Ilustra las alternativas descritas anteriormente para el desarrollo de software para el MCU.

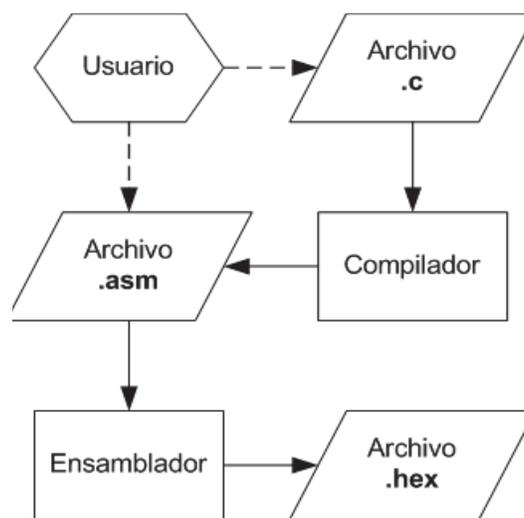


Figura 5.10. Desarrollo de software.

Una vez que se tiene el código ensamblado (.hex, .o, .bin, .coff) para el microcontrolador específico se utilizará un programa en la PC, el cual toma dicho código y lo envía mediante algún puerto (serial, paralelo, USB, etc.) a un dispositivo que lo escribe en la memoria del MCU (Figura 5.11).

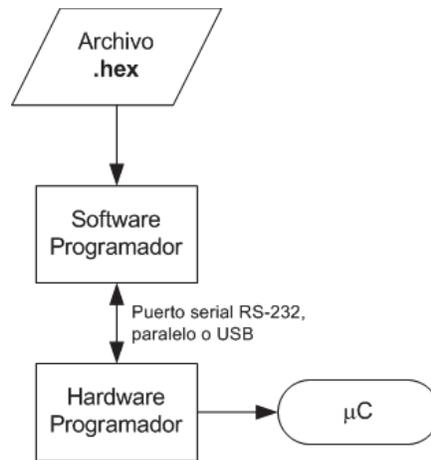


Figura 5.11. Proceso de programación del MCU.

5.5.1 Microcontrolador elegido

En el desarrollo del proyecto se decidió por la utilización del microcontrolador de 8 bits PIC18LF2550 de Microchip el cual presenta las siguientes características destacables para esta implementación:

- Soporte USB
- 32 Kb memoria de programa
- 2 Kb memoria RAM
- Voltaje de operación de **2 a 5.5 [V]**

Estas características se conjugan para lograr un dispositivo altamente eficiente en el uso de la memoria de datos, programa y por lo tanto en la velocidad de ejecución. Un punto importante acerca de la elección de este dispositivo se debió a que puede operar con un bajo voltaje (3.3 [V]) lo cual lo hace idóneo para operar en conjunto con el módulo XBee.

El modelo de la Figura 5.12, presenta uno de los varios encapsulados existentes, siendo este el más común, el DIP (Dual In-line package) por su facilidad en uso en una *protoboard*.

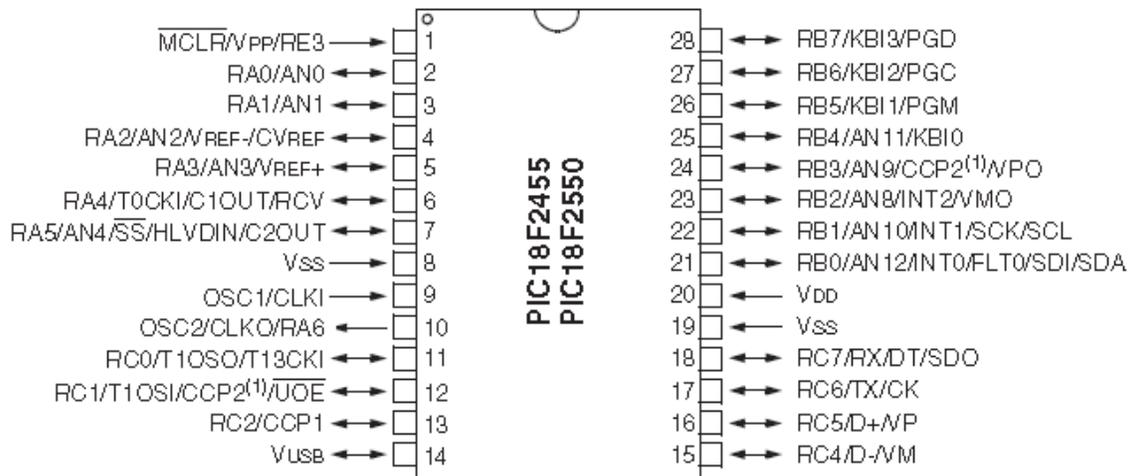


Figura 5.12. Microcontrolador.

5.6 Alcances y limitaciones

Debido al costo relativamente alto de los módulos XBee en el mercado, no es factible económicamente adquirir demasiados módulos. Por esta razón el sistema estará compuesto por: cinco módulos XBee-PRO; de los cuales tres son los encargados de realizar la localización de un cuarto módulo, que permitirá ubicar el objeto y por último el quinto XBee servirá como interfaz entre la computadora y la red 802.15.4 (red ZigBee). La visualización y adquisición de los datos en la computadora se hará por medio del puerto USB a través de la interfaz XBee Explorer USB la cual cuenta con una entrada USB mini dando acceso a los pines seriales y programables, permitiendo configurar el radio XBee.

La comunicación se realizó en una habitación, teniendo en cuenta las especificaciones técnicas de los módulos en cuanto alcance para ambientes interiores.

El diseño del dispositivo que portará el objeto a localizar y a este como tal, está fuera del alcance previsto del trabajo, por lo tanto el módulo que emite la señal no tendrá ningún encapsulamiento.

5.7 Desarrollo del proyecto

Después de realizar una descripción de los elementos necesarios para la realización de la aplicación correspondiente, procedemos a realizar una descripción detallada de todo el proceso de desarrollo de la aplicación.

El material, los dispositivos y herramientas utilizados en la implementación de la aplicación son los siguientes:

- PC con sistema operativo Windows 7 Profesional
 - Procesador AMD Turion 1.8 [GHz]
 - 2 [GB] en RAM
- XBee explorer USB
- Programador de Microcontroladores
- Multímetro
- Baterías
- Material para elaborar tarjetas de circuito impreso (cautín, placa fenólica, etc).
- Un puerto USB
- Un cable USB, para la comunicación entre el XBee explorer y la Computadora.
- 5 XBee-PRO
- 4 PIC18LF2550

Los lenguajes usados en esta aplicación son C para los PIC, y Java para la PC. El compilador C para PICs usado en este caso es el CCS por su inclusión de librerías que facilitan y disminuyen el tiempo de programación.

5.7.1 Funcionamiento del sistema

Existen tres entidades en cualquier sistema de localización: los dispositivos a localizar, un algoritmo de posicionamiento y un sistema de visualización. Los dispositivos o sensores miden alguna propiedad de la señal recibida de radio frecuencia que relaciona la distancia entre ellos y un punto de referencia fijo. El algoritmo de posicionamiento procesa las mediciones reportadas para estimar las coordenadas de la localización de los dispositivos. El sistema de visualización muestra la posición del dispositivo móvil al usuario. La Figura 5.13 muestra el diagrama de bloques de un sistema de posicionamiento.

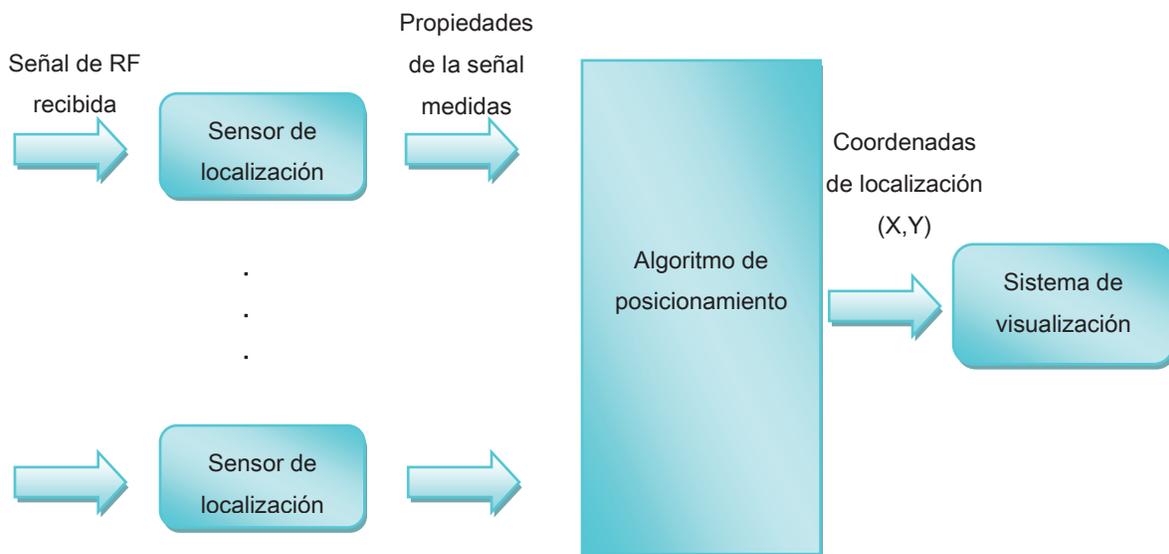


Figura 5.13. Diagrama a bloques.

El sistema de localización planteado consta básicamente de tres elementos: los módulos remotos, la infraestructura de red (dispositivos baliza) y el programa de localización (algoritmo de posicionamiento y el sistema de visualización).

Apoyándose en esos tres pilares, la implantación del sistema se realiza siguiendo un proceso que consta de dos fases: puesta en marcha mediante un entrenamiento del sistema, y la localización mediante la lectura de los valores de RSSI y el cálculo de la posición. Un diagrama del sistema se muestra en la Figura 5.14.

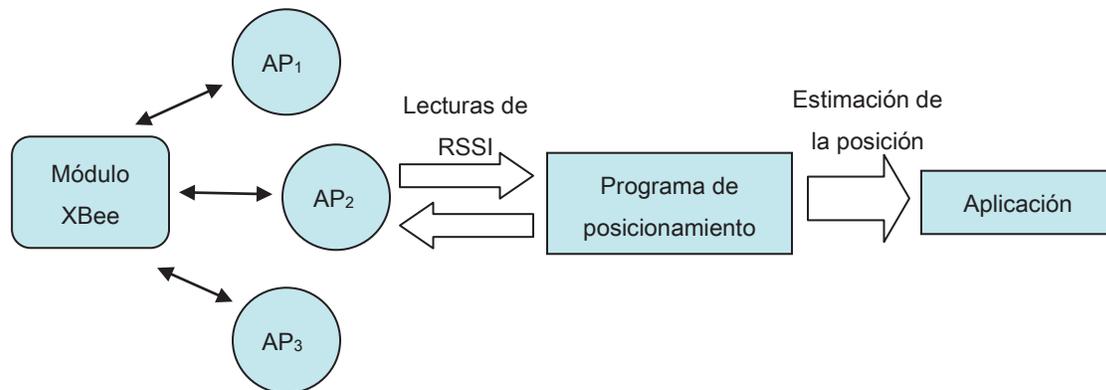


Figura 5.14. Sistema de localización.

Para la ubicación de los módulos, el sistema utiliza el método de reconocimiento de patrones o *fingerprinting*, que como se revisó en el Capítulo 3.2.2, consiste en generar un mapa de intensidad de señal recibida del área de interés, anotando el valor de intensidad de señal por cada punto de acceso para las diferentes posiciones en el mapa, esto constituye el entrenamiento del sistema. Debido a la naturaleza del método, únicamente sobre dichas posiciones medidas serán ubicados los módulos en el programa de localización.

El procesamiento de las señales se hará por medio de un microcontrolador que se programará debidamente para sacar la información asociada al RSSI.

Los módulos baliza miden los valores de potencia con respecto a los módulos remotos contenidos en la red y envían esta información a la computadora que contiene el algoritmo de posicionamiento que al momento de hacer la localización realiza una aproximación matemática buscando la posición con los valores de

potencia más parecidos a los obtenidos en el levantamiento inicial del sistema mediante el entrenamiento, acercando así la locación de los módulos hacia la posición más cercana de las obtenidas en la fase inicial. Finalmente las coordenadas de esta posición serán mostradas gráficamente en el sistema de visualización.

5.7.2 Infraestructura

La elección de la forma en que los dispositivos se comunicarán es importante para asegurar un correcto funcionamiento del conjunto. Un factor que determinará nuestra elección es el hecho de que se quiere conocer la posición de un objeto en todo momento, para ello será importante establecer unos puntos fijos que puedan servir de referencia a la hora de conocer la situación del dispositivo a localizar, así mismo estos deberán recoger datos del entorno.

La infraestructura consiste en una red inalámbrica ya establecida y debidamente configurada, sobre la cual los módulos inalámbricos (AP's o nodos baliza) se comunicarán con el software de la aplicación de localización contenida en una PC, a fin de ubicar dicho módulo sobre un plano de la zona en la cual se va a trabajar.

Dicha infraestructura está formada por una red ZigBee 802.15.4 y una computadora que centraliza las operaciones del software de localización. La red debe disponer de una densidad de nodos baliza que permita un posicionamiento preciso, de al menos tres nodos para realizar la localización tomando en cuenta que los módulos remotos deben estar dentro del área de cobertura de ellos.

El sistema consta de tres puntos de acceso como se aprecia en la Figura 5.15.

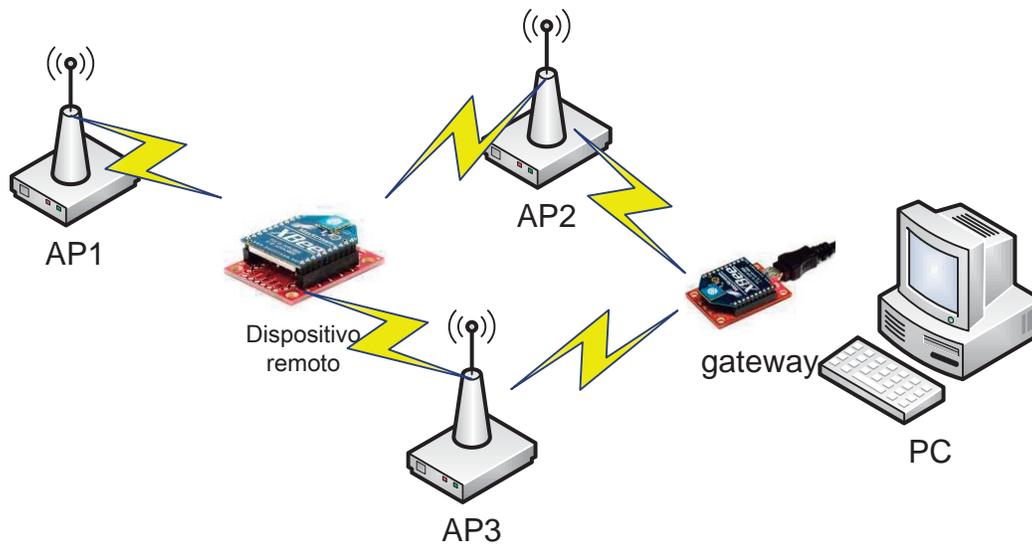


Figura 5.15. Infraestructura de la red ZigBee.

La comunicación que se establezca entre el móvil y los dispositivos fijos (nodos baliza) será la que nos permita obtener la información requerida para determinar la posición buscada.

Dentro de la red se debe incluir una computadora central encargada de la recolección de la información, así como del correcto análisis de esta.

El prototipo del sistema propuesto consta de tres elementos:

- El módulo remoto
- Los nodos baliza o dispositivos localizados en una posición fija y
- El módulo que se encuentra conectado a la computadora (Gateway)

Todas las partes involucradas no contienen ningún tipo de interconexión física ya que se comunican vía inalámbrica, sin embargo el sistema depende de cada una de las partes para su funcionalidad completa.

5.7.3 Módulo remoto

Los módulos remotos o dispositivos a localizar, son dispositivos compactos que notifican su posición al servidor de localización cada vez que se lo solicitan, esto a través de la comunicación con los nodos baliza. Son dispositivos autónomos, es decir, cuentan con su propia fuente de alimentación que consiste en una batería. Estos posibilitan la localización de cualquier persona o dispositivo al que se adjunten, dentro de la cobertura de la red. Se ha tratado que tengan un tamaño moderadamente reducido por la característica de portabilidad que se requiere en el sistema.

El componente principal del módulo remoto lo constituye el módulo inalámbrico XBee al cual se le ha adicionado un microcontrolador con cierto grado de “inteligencia” con la capacidad de reconocer tramas API, su interconexión es a través de una interfaz UART. A estos dos elementos se le han agregado componentes para conformar el módulo remoto.

La tarjeta electrónica cuenta con conectores y sócalos para la alimentación con baterías, además de sus respectivos filtros, cristal oscilador y leds de señalización para indicar el estado del módulo.

Está encargado de leer todas las tramas que le son enviadas y retransmitirlas a las balizas distribuidas en el interior de la construcción (operación que es comandada y dirigida por el microcontrolador). Al enviar tramas en formato API, las balizas podrán leer el valor RSSI que será útil para poder estimar la posición del objeto móvil.

La Figura 5.16 muestra el diagrama de flujo que indica la operación del programa del dispositivo inalámbrico.

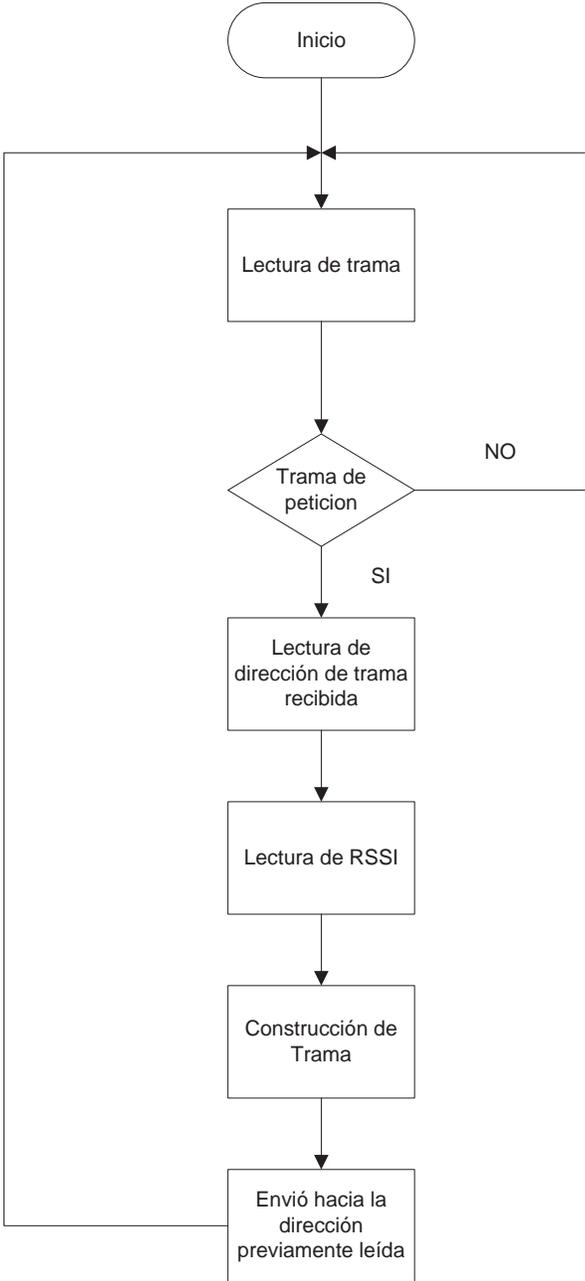


Figura 5.16. Diagrama de flujo del dispositivo remoto.

El dispositivo implementado, trabaja únicamente con cuatro de sus 20 pines disponibles del Módulo XBee; estos son: los pines de transmisión (DI), recepción (DO), tierra (GND), y el pin de alimentación a 3.3 [V] (VCC).

Una de las principales ventajas del módulo XBee es su interfaz serie UART. Esta interfaz hace que sea ideal para la comunicación con un microcontrolador PIC. El XBee se conecta directamente al microcontrolador por medio del uso de dos de sus pines: de transmisión y el de recepción UART, lo cual es posible debido a que entre el módulo XBee y el microcontrolador no existe proceso de conversión de voltaje ya que ambos chips operan a 3.3 [V], sólo es posible esto gracias a que el microcontrolador ocupado es el PIC18LF2550 el cual puede operar a baja tensión (3.3 [V]). Sin embargo, hay que tomar en cuenta que los pines transmisor y receptor del Módulo XBee, se conectan de manera cruzada a los pines de transmisión y recepción del microcontrolador, para que mientras el uno transmite, el otro reciba los datos y viceversa. La Figura 5.18 muestra la conexión existente entre el módulo XBee y el microcontrolador.

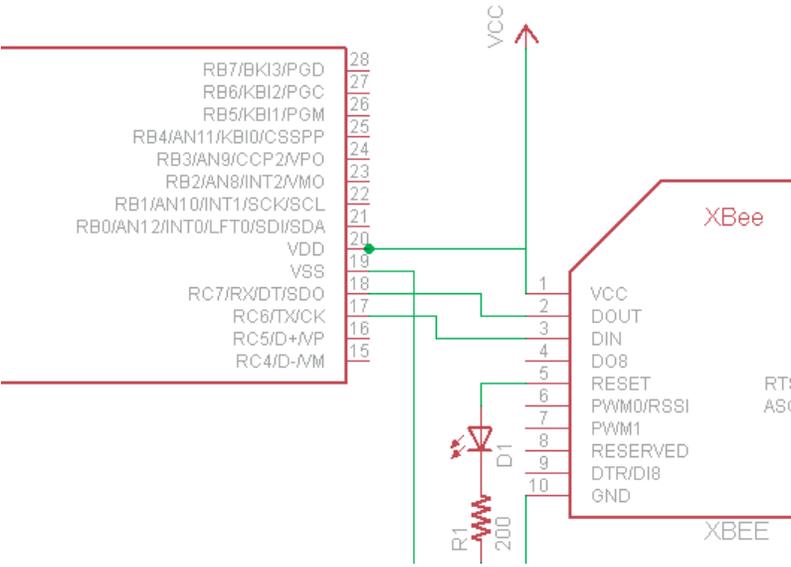


Figura 5.18. Conexión UART entre el módulo XBee y el microcontrolador.

Para el correcto funcionamiento del microcontrolador se activa conectando el pin1 del PIC que es el MCLR a VCC. Para el modo de oscilación se utiliza el modo HS, un cristal de 4 [MHz] que es conectado a los pines OSC1 y OSC2 para establecer una oscilación. La Figura 5.19 muestra la conexión de estos pines a un oscilador de cristal.

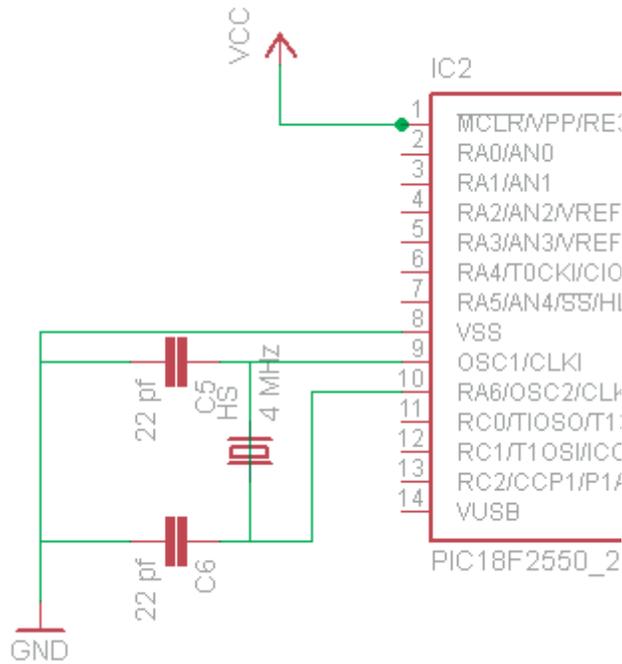


Figura 5.19. Configuración del oscilador de cristal.

En la Figura 5.20 se presenta el diagrama esquemático final del dispositivo probado e implementado, y en la parte de Anexos se encuentra la gráfica del ruteo perteneciente al diagrama dispositivo inalámbrico.

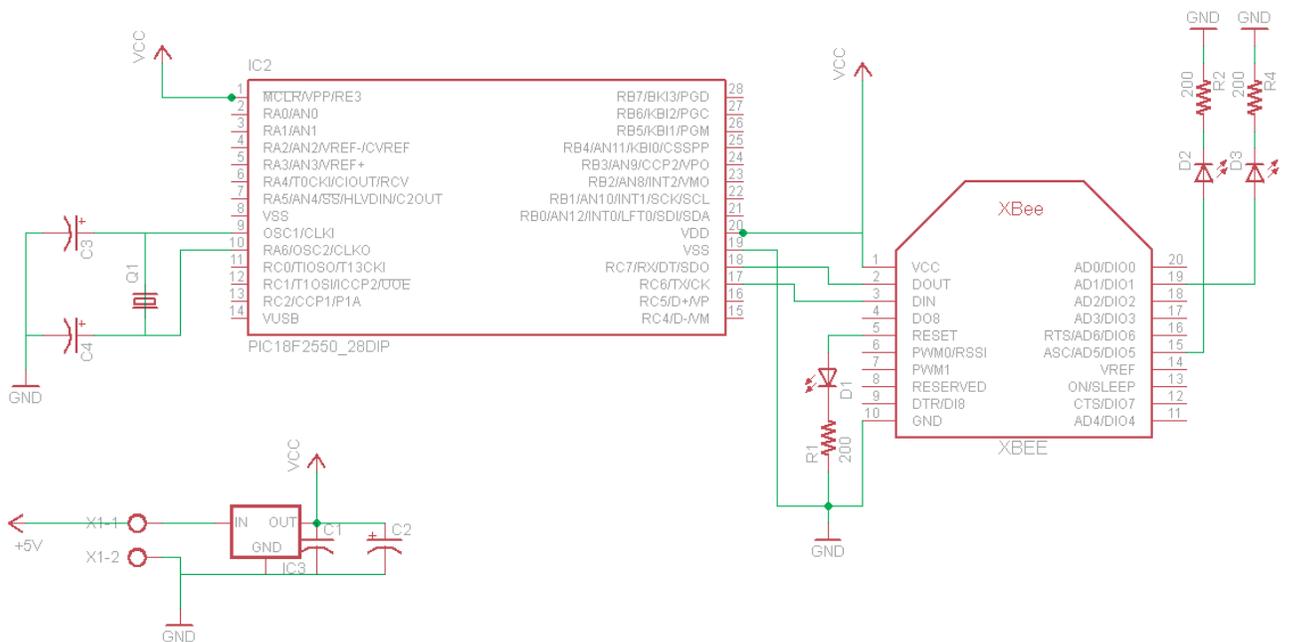


Figura 5.20. Esquemático del dispositivo remoto.

5.7.4 Nodos Balizas

Las balizas tienen como misión recoger el valor RSSI correspondiente al dispositivo móvil para posteriormente enviar dicha información a la computadora y así comenzar con el procesado de datos. Estos nodos se encuentran situados en posiciones fijas en el interior de la construcción por lo que están alimentados por la red eléctrica.

Al igual que en la explicación del Dispositivo móvil, este incluye un módulo XBee conectado a los pines de transmisión y recepción de la interfaz UART de un microcontrolador PIC18F2550, así como sus correspondientes pines de alimentación. Sin embargo, al ser difícil la adquisición de microcontroladores de bajo consumo, se implementó el mismo circuito, pero ahora tomando en cuenta la diferencia de voltajes entre el módulo XBee y el microcontrolador PIC18F2550, el cual debe ser alimentado con 5 [V].

Las conexiones entre el microcontrolador y el módulo XBee son directas, pero es muy importante tomar en cuenta los enlaces entre los pines TX/RX. Entre el pin de salida (TX) del microcontrolador y el pin de entrada del XBee (DI), tiene que implementarse necesariamente un divisor de voltaje ya que el nivel de tensión de entrada del módulo no debe de exceder los 3.3 [V], esto se logró reduciendo el voltaje por medio de resistencias conectadas en serie para obtener 2/3 del voltaje de salida del MCU.

$$5[V] \left(\frac{2}{3}\right) = 3.3[V]$$

La Figura 5.21 muestra la configuración que debe existir entre el módulo y el Microcontrolador.

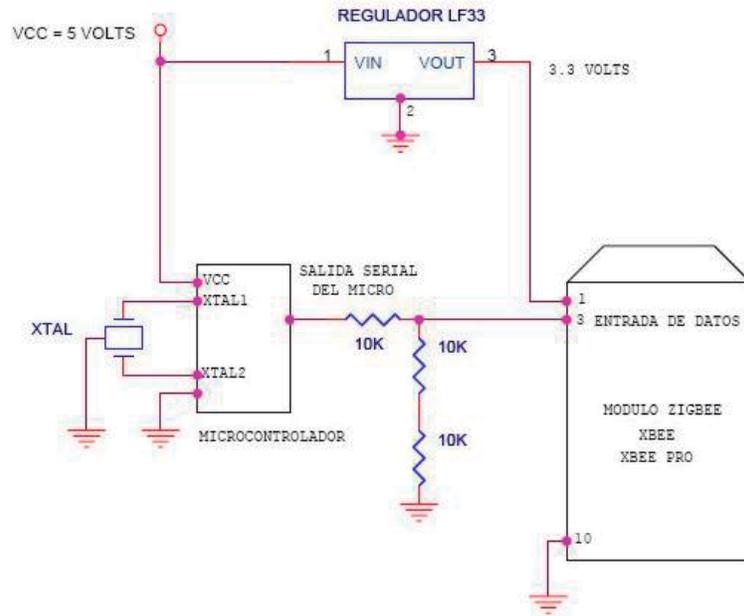


Figura 5.21. Conexión para la transmisión.

Con respecto a la conexión entre el pin de entrada (RX) del microcontrolador y el pin de salida del módulo XBee (DO), la conexión es directa sin implementar nada entre ambos pines, esto gracias a que una de las características que posee el microcontrolador es que puede manejar el nivel de tensión que entrega el módulo XBee al momento de la recepción. La Figura 5.22 muestra la conexión que debe existir entre ambos dispositivos.

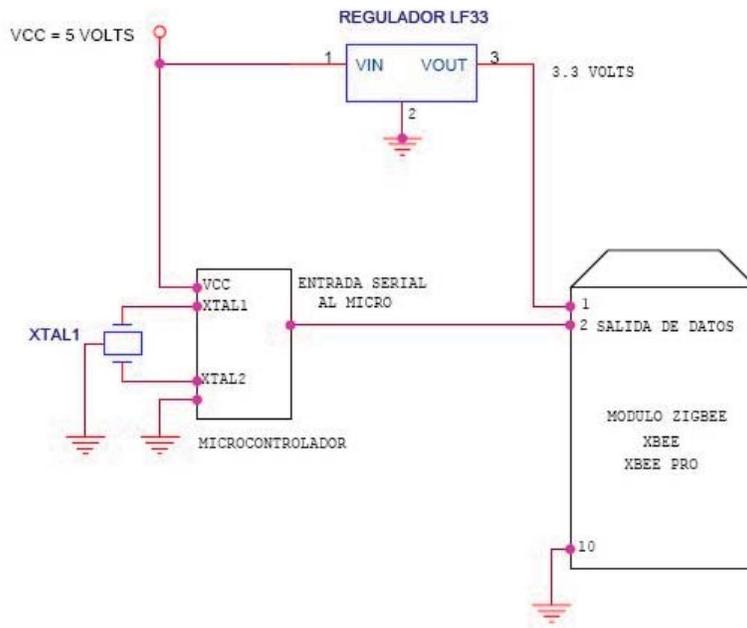


Figura 5.22. Conexión para la recepción.

La placa toma corriente de un eliminador de 12 [V], cuyo voltaje pasa primeramente por el regulador 7805 para que alimente a todo el circuito con 5 [V], excepto al Módulo XBee, debido a que este trabaja exclusivamente con 3.3 [V], por tanto se necesita un regulador LF33CV para obtener de la misma fuente de alimentación el voltaje requerido. La regulación de voltaje de 5 [V] permite al microcontrolador trabajar correctamente además de ofrecer una protección contra voltajes inversos ocasionados en el momento de encender el dispositivo.

De manera general, se puede decir que los nodos balizas esperan a recibir una transmisión desde el dispositivo móvil o desde la computadora, al recibir una transmisión correcta se procede a identificar qué tipo de trama es, en caso de que provenga de la computadora, ésta es interpretada como una orden para hacer una petición hacia la dirección que venga en el contenido de dicha trama, en cambio en caso de la trama provenga del nodo móvil se procede a leer el valor RSSI de la trama recibida y se crea una nueva trama con la dicha información para ser enviada hacia la PC. De manera general, la Figura 5.23 muestra el diagrama de flujo de la operación del programa del nodo baliza.

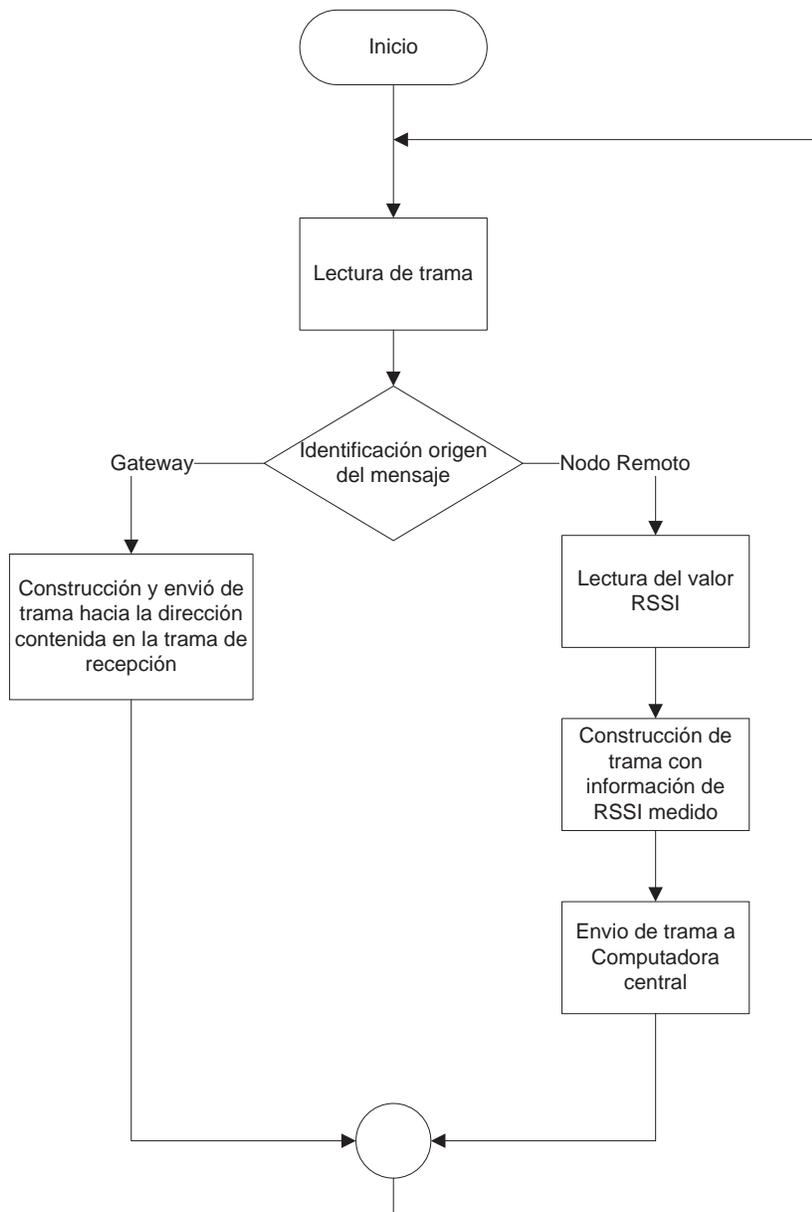


Figura 5.23. Diagrama de flujo, Nodo Baliza.

5.7.5 Hardware en la computadora receptora

Una vez que se tienen los emisores listos para el envío del mensaje, se implementará el lector para poder recibir todas las tramas y procesarlas.

La computadora receptora es el “cerebro” de la red, ésta controla y permite la comunicación con los dispositivos que pertenecen a la red ZigBee. El funcionamiento constante de este dispositivo es imprescindible ya que estará permanentemente recibiendo información por parte de los nodos baliza, así como éste servirá para configurar los demás elementos de la red, una caída de este elemento de la red es fatal para el funcionamiento del sistema así como también un mal funcionamiento del hardware de comunicaciones a él asociado.

Los módulos XBee pueden trabajar de dos maneras: se los puede adaptar a un circuito tomando en cuenta aspectos de polarización ya que los módulos operan con 3.3 [V] necesariamente, y una segunda forma es haciendo uso de tarjetas de desarrollo y exploradores, como para este caso en el cual la computadora se comunica con el XBee utilizando el XBee Explorer USB.

El XBee Explorer USB contiene receptáculos de 20 pines para ubicar el módulo, y éste cumple funciones como la de proveer de voltajes de polarización necesarios para el correcto uso de los módulos, y convierte los voltajes RS232 que entrega la computadora a niveles de voltaje de 3.3 [V] tanto para la transmisión como para la recepción de datos durante la comunicación.

Para mantener un diseño sencillo se optó por implementar el Gateway receptor en una única computadora, el cual consta de una PC conectado mediante una interfaz USB hacia el módulo XBee, éste asumirá el papel de Coordinador de la red ZigBee, y se comunicará con los módulos remotos mediante tramas API.

El Gateway se conecta vía USB a la computadora utilizando el XBee Explorer USB comportándose como un puerto serie virtual, esto requerirá de la instalación de un driver FTDI para el correcto funcionamiento de la tarjeta exploradora.

En la conexión de la Computadora con la red ZigBee se encuentran involucrados la PC, el explorador USB y el módulo XBee, cuya adaptación se la muestra a continuación en la Figura 5.24.



Figura 5.24. Implementación del dispositivo receptor.

La PC, por medio de un software desarrollado especialmente para la aplicación, es la que administra y almacena los datos provenientes de los nodos remotos.

5.7.6 Software de configuración

Los módulos XBee son provistos de una herramienta en software para la configuración de los parámetros y actualización del firmware llamada X-CTU, que tiene la capacidad de realizar las siguientes tareas:

- Actualización del firmware en un módulo local, para esto se quiere una conexión USB o Serial.
- Leer o escribir los parámetros de configuración de un dispositivo local o remoto.
- Guardar o cargar los perfiles de configuración personalizados.

Cuando inicia, el X-CTU se puede ver cuatro pestañas en la parte superior del programa (ver Figura 5.25). Cada una de estas pestañas tiene una función diferente. Las cuatro pestañas son las siguientes:

- PC Settings. Permite al usuario seleccionar el puerto serial COM y configurarlo para acceder de los módulos.
- Range Test. Permite al usuario realizar una prueba de intensidad de señal recibida entre dos módulos.
- Terminal. Permite el acceso al puerto serial COM. Además, esta pestaña permite configurar al módulo por medio de comandos AT. Los comandos AT se encuentran detallados en las hojas técnicas de los módulos XBee.
- Modem Configuration. Permite modificar los parámetros del firmware cargado en el módulo a través de una interfaz gráfica de usuario. Esta pestaña permite al usuario cambiar las versiones del firmware.

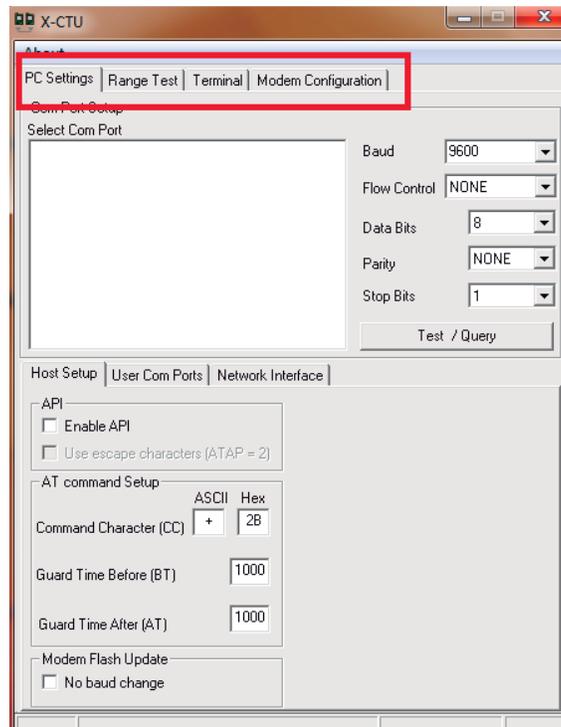


Figura 5.25. Software X-CTU.

5.7.7 Arquitectura de la red

La red está formada por 5 dispositivos siendo uno de ellos el principal que recibe la información de cada uno de los otros. Se distinguen los siguientes componentes:

- Dispositivo móvil o dispositivo remoto. Será el dispositivo que lleva la persona u objeto a localizar. Está alimentado con baterías y se mantiene permanentemente en espera de recibir peticiones por parte de los nodos baliza y responder inmediatamente esta petición de tal manera que los nodos baliza puedan medir el RSSI del enlace.
- Nodos Baliza. Es un dispositivo igual que el móvil sólo que está alimentado por la red eléctrica. Los nodos balizas se encuentran colocados en posiciones fijas dentro de la construcción. Aparte de recibir datos del dispositivo móvil, deben encargarse de medir la intensidad de la señal que

recibe por parte de los nodos móviles y enviarla dicho dato a la computadora central para su posterior análisis.

- Computadora central. Es el último dispositivo de la red y está encargado de recoger los datos que se comparten y generan en toda la red. Como se ha comentado, es el encargado de almacenar la información y procesarla.

La Tabla 5-4 muestra un resumen de los dispositivos incluidos en la red.

Tabla 5-4. Dispositivos de la red.

Dispositivo	Alimentación	Posición	Función
Dispositivo móvil	Batería	Móvil	Notificar posición
Baliza 1	Red eléctrica	Fija	Recoger datos de los nodos móviles
Baliza 2	Red eléctrica	Fija	Recoger datos de los nodos móviles
Baliza 3	Red eléctrica	Fija	Recoger datos de los nodos móviles
Computadora central	Red eléctrica	Fija	Almacenar y procesar la información recibida por parte de los nodos baliza

Una vez que indicados los componentes de la red se requiere hacer una configuración especial en cada uno de los módulos XBee para que estos puedan intercambiar información entre sí. Los parámetros que entran en juego son:

- ID: La identificación de red sobre la que se va a trabajar. Cada uno de los módulos debe tener el mismo ID.
- C: Canal por el que se transmite. Todos los módulos tienen el mismo valor de C.
- CE: Configuración para trabajar como “End Device”. Todos los dispositivos deberán tener este valor a 0.
- A1: Asociación de “End Devices”. Todos los dispositivos deberán deshabilitar esta función escribiendo un 0 en este campo.
- DH y DL: Direcciones de destino alta y baja, permite dirigir paquetes a un único módulo.
 - DH: Deberá configurarse en todos los dispositivos de la red como 0.

- DL: Envía los datos al dispositivo cuyo MY coincida con el DL del actual. Deberá configurarse en cada dispositivo de manera especial como se ve en la Tabla 5-5.
- MY: Dirección fuente del módulo XBee, es decir, de quién vamos a aceptar la información. Deberá configurarse en cada módulo de manera especial como se ve en la Tabla 5-5.

La Tabla 5-5 indica la configuración que se ha hecho para cada uno de los dispositivos de la red.

Tabla 5-5. Configuración de los módulos.

Parámetros	Disp. móvil	Baliza 1	Baliza 2	Baliza 3	Comp. central
ID			1AAA		
C			C		
CE	0	0	0	0	0
A1	0	0	0	0	0
DH	0	0	0	0	0
DL	5000	5000	5000	5000	0
MY	5003	5001	5002	5004	0

Siguiendo la configuración anterior, se tiene completa la red, permitiendo la transmisión correcta entre los dispositivos que la conforman.

De forma muy general, se puede decir que según el grado de atenuación que perciba el módulo remoto en los datos recibidos procedentes de los nodos baliza, podrá estimar su cercanía o lejanía de éstos y así hacerse de una inferencia de su localización respecto de ese punto fijo de la red.

La Figura 5.26 muestra un esquema del modo en que un usuario u objeto se puede localizar. En la misma se observa que el dispositivo móvil recibe las señales de n puntos distintos, obteniendo un vector de potencias con n elementos. Utilizando como punto de partida este vector, una computadora calcula la posición del objeto dentro de la edificación.

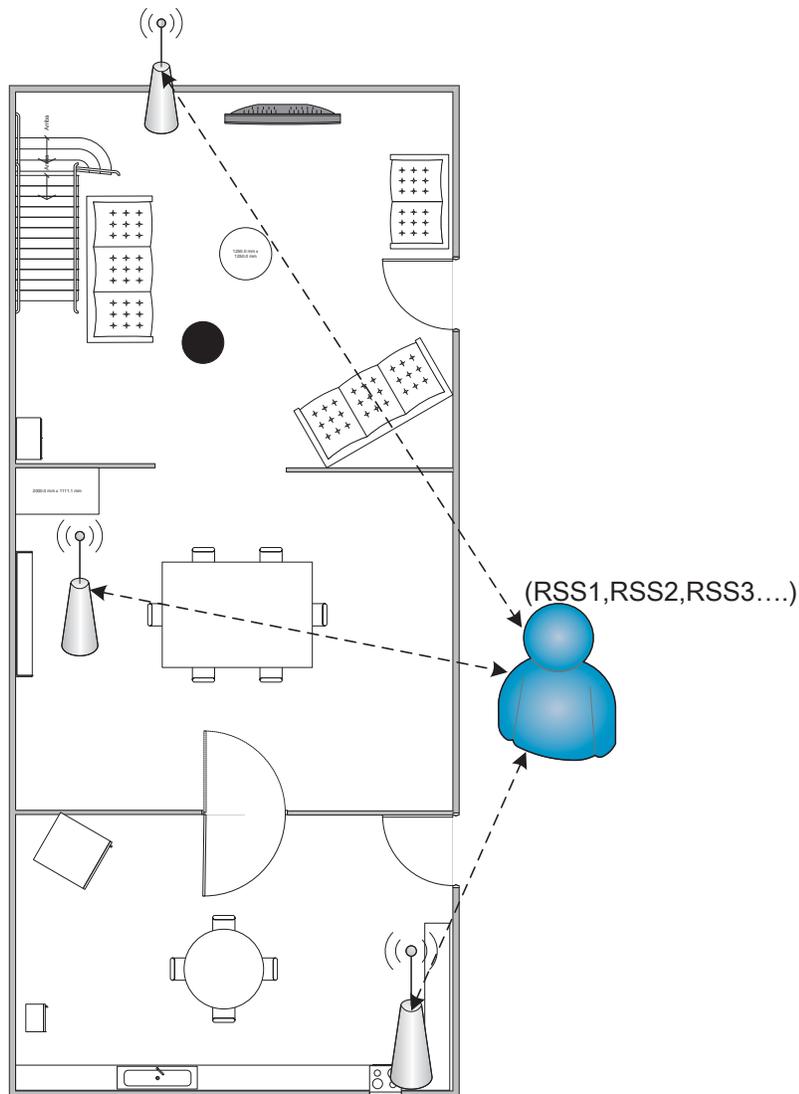


Figura 5.26. Esquema general de la aplicación.

A continuación se realiza una exposición detallada de los fundamentos teóricos del sistema de localización, y finalmente se describirá el software desarrollado así como el algoritmo implementado.

6. Diseño del software

La elección del lenguaje de programación utilizado a la hora de implementar una aplicación, dentro de un amplio espectro de posibilidades, queda condicionada por el tipo de aplicación a desarrollar.

Los lenguajes de propósito específico están enfocados a desarrollar una clase concreta de aplicaciones del modo más sencillo posible, estando su semántica restringida al propósito para el que fueron creados. Por otro lado, los denominados lenguajes de propósito general tienen como objetivo el poder representar la solución de cualquier tipo de problema computacional.

Lenguajes como Java y C++, ambos de propósito general, orientados a objetos y de sintaxis análoga, pueden ser elegidos, por ejemplo, para desarrollar aplicaciones portables y distribuidas en el caso de la primera, o bien para crear aquéllas en las que su eficiencia en tiempo de ejecución es primordial para el caso de C++.

Por otra parte en la elección del medio de programación a utilizar es importante considerar el alto costo de las licencias de los lenguajes de programación, punto crucial para este proyecto puesto que es lo que hasta cierto punto se pretende evitar.

6.1 Lenguaje de programación.

Al momento de optar por tener una interfaz para el usuario, se debe elegir el lenguaje para programar la misma. De la gran gama de lenguajes existentes hoy en día se decidió optar por la utilización de Java como lenguaje de programación para el desarrollo de la aplicación, y a continuación detallan los motivos de la dicha elección:

Como primer punto se tiene que Java cuenta con una interfaz diseñada específicamente para la comunicación con los XBee/XBee-Pro series 1 (802.15.4) y series 2 (ZNet 2.5 and ZB/ZigBee Pro) OEM RF Modules. Particularmente el API (Application Programming Interface) utilizado para la comunicación es: `com.rapplogic.xbee`. Cabe destacar que para desarrollar el software fue necesario instalar el JDK en el equipo que se utilizó para programar, además de un IDE (Spring Source Tools). Todas las instalaciones se realizaron según los procedimientos estándar de instalación indicados en las páginas de los desarrolladores.

Otro punto clave y uno de los más importantes, es que Java es de desarrollo abierto (Open Source), por lo no se depende de una cierta compañía para el soporte y *debugging*, ni tampoco se debe pagar una licencia para el desarrollo, característica que se consideró fundamental en el desarrollo del proyecto para reducir en lo posible los costos.

Así mismo, al ser Open Source, se tiene una gran comunidad activa de desarrolladores que permite encontrar una solución rápida a cualquier traba que se interponga a la hora de programar. Esta comunidad también actualiza y desarrolla nuevas librerías por lo que se puede encontrar una gran variedad de funciones a utilizar, facilitando así el desarrollo de la aplicación.

Además de los motivos antes expuestos existen otras ventajas adicionales como su portabilidad, que permitiría ejecutar el software en computadoras con diferentes plataformas y sistemas operativos sin necesidad de reprogramarla aplicación, en el caso de este proyecto esto no es estrictamente necesario ya que se decidió realizar el desarrollo sobre Windows, pero deja la posibilidad de migrarlo en un futuro a otro sistema operativo.

Adicionalmente existe una gran cantidad de conocimiento y recursos sobre Java, ya sea en la web, foros, libros, etc.

Por último y no por eso menos importante, se contempló que la realización del proyecto tiene muchos desafíos y puntos de investigación, se intentó buscar entre las opciones posibles, aquella que generara menor incertidumbre al momento de la implementación. En el caso de Java, el autor se encontraba familiarizado con el lenguaje, debido a que durante su formación académica ha desarrollado software para la Facultad de Ingeniería a través de la Unidad de Servicios de Cómputo Académico (UNICA), lo que le aportó las herramientas necesarias para el desarrollo del programa sin necesidad de invertir tiempo extra en investigación y aprendizaje de algún otro lenguaje de programación.

En las siguientes partes de esta sección se detallan los puntos más importantes referentes al desarrollo del software. Pero primero se hará una breve reseña sobre Java.

6.2 Java

Java es un lenguaje de programación orientado a objetos desarrollado originalmente por James Gosling y sus colegas en la empresa Sun Microsystems a principios de los años 90.

Fue concebido con los siguientes objetivos:

- Usar la metodología de programación orientada a objetos
- Permitir al mismo programa ejecutar en diferentes plataformas de computación
- Soporte interno para el uso en redes de computadoras
- Ejecutar en forma segura código residente en máquinas remotas

Un programa en el lenguaje Java se compila para ser ejecutado sobre una plataforma independiente. Esta plataforma de ejecución independiente está formada básicamente por:

- La máquina virtual de Java (*Java Virtual Machine*).
- La interfaz de programación de aplicaciones en Java o core API (*Application Programming Interface*).

Esta dualidad, que consigue la independencia de una plataforma y el mayor nivel de abstracción en la programación de aplicaciones. El API es un conjunto de clases que están compiladas en el formato de código binario de la máquina abstracta. Estas clases son utilizadas por el programador para realizar aplicaciones de una forma más sencilla.

La mayoría de los lenguajes de programación se caracterizan por ser interpretados o compilados, lo que determina la manera en cómo serán ejecutados en una computadora. Java tiene la característica de ser al mismo tiempo compilado e interpretado.

El compilador es el encargado de convertir el código fuente de un programa en un código intermedio llamado *bytecode* que es independiente de la plataforma en que se trabaje y que es ejecutado por el intérprete de Java que forma parte de la Máquina Virtual de Java.

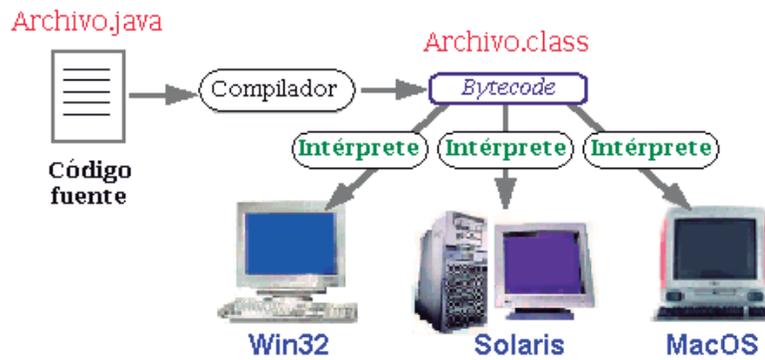


Figura 6.1. Compilación y ejecución de un programa java.

Es por eso que se dice que Java tiene la característica de ser multiplataforma, pues el mismo *bytecode* (o archivo *.class), es interpretado por una máquina virtual diferente y adecuada para cada plataforma, es decir, que lo que cambia no es el archivo compilado, si no la máquina virtual que lo va a interpretar.

6.3 Algoritmo

Parte fundamental del sistema de localización lo constituye el algoritmo de posicionamiento, pues es el encargado de realizar el cálculo para determinar la posición del módulo remoto. Este cálculo lo hace en base a los valores de potencia recibidos por parte de cada uno de los nodos baliza, medidos y enviados a la computadora central.

En los últimos años se han desarrollado varios tipos de algoritmos basados en el reconocimiento de patrones. Uno de ellos es el algoritmo KNN (*K-Nearest Neighbors*) que se basa en el recubrimiento de la planta del edificio con nodos baliza, con el requisito de que sus huellas de potencia se solapen unas con otras.

La idea sobre la que se basa el algoritmo es que, aunque RSSI recibido en el módulo móvil no depende linealmente de la distancia al nodo baliza, sí que guarda alguna relación; al menos se puede asegurar que si el nodo baliza está muy cerca

del terminal móvil la potencia recibida será alta, y si el nodo baliza está lejos, el RSSI disminuirá. Mediante el indicador de potencia de la señal recibida (RSSI) de cada nodo baliza en un punto, se puede inferir la posición. La Figura 6.2 ejemplifica el principio en el cual se basa el algoritmo.

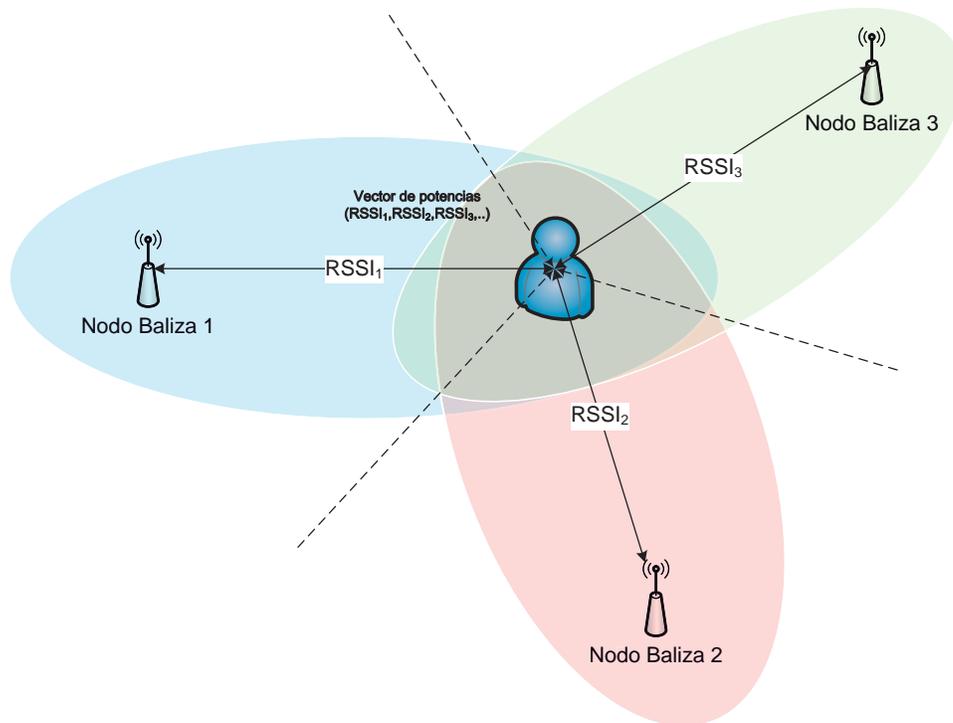


Figura 6.2. Recepción del vector de potencia.

Con base a esta simple idea, se crea un radio mapa, que contiene una base de datos de localizaciones (Tabla 6-1), que incluye la posición en el espacio de cada punto y el RSSI asociado a cada uno de los nodos baliza.

Tabla 6-1. Tabla generada en el levantamiento.

Punto de Localización	X [m]	Y [m]	Z [m]	RSSI ₁ [dBm]	RSSI ₂ [dBm]	...	RSSI _m [dBm]
Punto 1	X ₁	Y ₁	Z ₁	RSSI _{baliza1,1}	RSSI _{baliza1,2}	...	RSSI _{baliza1,m}
Punto 2	X ₂	Y ₂	Z ₂	RSSI _{baliza2,1}	RSSI _{baliza2,2}	...	RSSI _{baliza2,m}
Punto 3	X ₃	Y ₃	Z ₃	RSSI _{baliza3,1}	RSSI _{baliza3,2}	...	RSSI _{baliza3,m}
⋮	⋮	⋮	⋮	⋮	⋮	...	⋮
Punto n	X _n	Y _n	Z _n	RSSI _{balizan,1}	RSSI _{balizan,2}	...	RSSI _{balizan,m}

Donde X , Y , Z son las posiciones en el espacio y n será el número de puntos que se muestrearán y se almacenarán en la tabla. El número total de nodos baliza será m . Por tanto, el indicador de potencia $RSSI_{baliza,n,m}$ denota la potencia recibida del nodo baliza m en el punto n .

Hay que tener en cuenta que las mediciones del radio mapa serán válidas siempre que no cambie la posición de ninguno de los nodos baliza o se introduzcan nuevos objetos u otros agentes que puedan atenuar la señal.

Es evidente que las variaciones de la potencia de señal también son un inconveniente grave para los métodos de reconocimiento de patrones. Sin embargo, en éstos últimos, existe la posibilidad de encajar una muestra dentro de un conjunto sin que haya certeza absoluta de la pertenencia al mismo. Gracias a ello, podremos llegar a distinguir entre distintas localizaciones siendo métodos más inmunes a la variabilidad inherente de la señal inalámbrica.

Una vez que se quiere localizar el módulo remoto, se obtiene un vector de potencias x en la posición a calcular, el cual se compara con la matriz de posiciones obtenida en la fase anterior.

El algoritmo KNN, trata de obtener los K puntos de la matriz de posiciones más cercanos al vector de potencia medido. Una vez obtenidos, se promedian y con ello se obtiene la posición.

El criterio usado para seleccionar el punto más probable de localización de un dispositivo es la menor distancia euclídea, tomando los valores de RSSI como distancias métricas. Y así hallaremos el punto donde se encuentra el cliente.

El algoritmo, detallado en pseudocódigo, sería así:

```
ptos_escaneo = vector de fuerzas obtenidas en el escaneo actual
data = base de datos creada durante el levantamiento (fingerprinting)
points = vector con las fuerzas obtenidas en cada punto
```

Estimación de la posición:

```
para cada  $p_i \in \text{points}$ : distancia(ptos_escaneo,  $p_i$ )
    si distancia es la menor, ese es el punto buscado
```

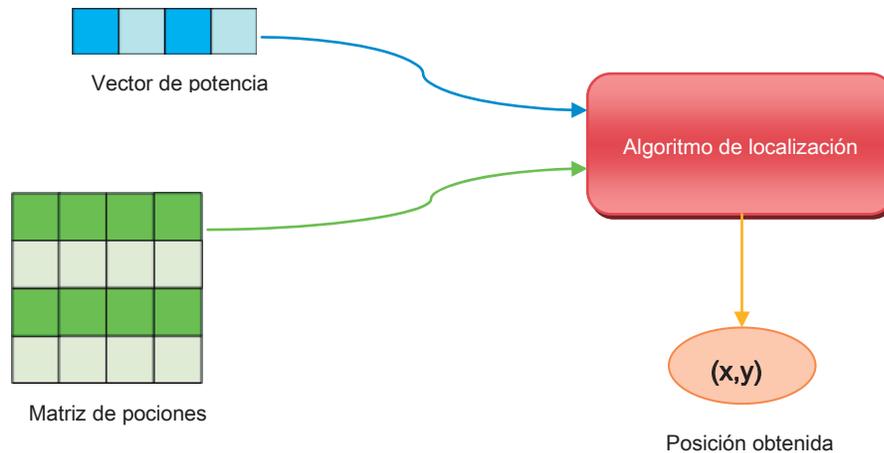
Donde distancia(ptos_escaneo, p_i):

$$d(p, p_i) = \frac{1}{M} * \sqrt{\sum_{j=1}^M (RSSI_j(x, y) - RSSI_j(x_i, y_i))^2}$$

donde M = número de elementos de ptos_escaneo

donde $RSSI_j$ = fuerza de la señal recibido del nodo baliza_j en (x, y)

Este método tiene la ventaja de que es muy fácil de implementar y no requiere cálculos complejos. Obtiene la posición desde la primera medición y no es necesario un período de arranque hasta que empiece a proporcionar valores válidos. El problema que puede surgir es que cuanto mayor sea la matriz de posiciones; tanto por la longitud de los vectores de potencia (que depende de los diferentes nodos balizas almacenados), como por el número de posiciones almacenadas; mayor será la complejidad del método.



Otro punto importante que se debe considerar es que, durante el proceso de *fingerprinting*, al tomar valores de señal en un punto, se toman varias lecturas de la señal en el mismo lugar, con lo cual se obtienen varios valores de RSSI de los diferentes nodos baliza registrados en el sistema en un mismo punto, así que para que el algoritmo sea eficiente, almacenamos la media de dichas lecturas, y ese es el valor que se tiene en cuenta al hallar las distancias.

Así mismo este algoritmo es lo suficientemente ligero para ser ejecutado con rapidez en cualquier computadora. Esto consigue que el proceso de localización se pueda ejecutar en dispositivos con capacidad computacional reducida.

6.4 Interfaz

Una de las principales funciones del sistema es la coordinación de los módulos XBee conectados a la red o PAN. Esto se consigue, desde el punto de vista físico accediendo a la red mediante un transceptor inalámbrico. Para ello se ha utilizado el XBee Explorer el cual actúa como un adaptador serie-usb para poder conectar el XBee con la PC a través de un puerto USB.

De esta forma, toda la información que el XBee envíe a través de su bloque de comunicaciones serie (interfaz uart), será recibida en la PC y estará accesible para la aplicación.

La aplicación del sistema, está implementada sobre un sistema operativo Windows, usando el lenguaje de programación Java. Se hace imprescindible el uso de una librería específica que realice dos funciones esenciales: hacer de interfaz entre el XBee y nuestra aplicación y que permita manejar las tramas estructuradas de datos que maneja el XBee, a nivel de objetos.

Para lograr todo esto se ha utilizado la única librería libre y open source que hasta este momento existe, XBee-API (<http://code.google.com/p/XBee-api/>), que implementa la mayoría de la funcionalidad del XBee, en lenguaje Java. Internamente se apoya en la librería también libre y open source, RXTX que da soporte de bajo nivel para el acceso a los puertos USB en lenguaje Java.

Una vez conectado el XBee a través del XBee Explorer al puerto USB, dinámicamente el sistema operativo reconoce el conjunto como un dispositivo con comunicaciones serie y lo hace accesible a través de un puerto COM.

6.5 Gestión de tramas

Todas las tramas que llegan al coordinador, son enviadas a través del puerto USB, generando una interrupción en el sistema operativo. Esta interrupción es propagada hasta la máquina virtual java (JVM), de tal forma que el núcleo de la librería RXTX es notificado del hecho de la interrupción y la atiende, recogiendo los datos sin un formato específico como una cadena de bytes. Estos bytes son nuevamente recogidos por la librería XBee-API, que los encapsula en las estructuras de tramas que marca la especificación XBee, además de encapsular la información de forma estructurada, provee de métodos y funciones para manejar

con cierta facilidad a alto nivel dichas tramas. Con base en lo anterior se cuenta con la capacidad de administrar, por medio de software, las tramas que circulan por la red. Ahora bien, como se requiere estar monitoreando la red 802.15.4 continuamente hay que tener en cuenta los posibles cuellos de botella que existen desde que el XBee recibe la información hasta que ésta es procesada y almacenada.

Cada vez que se recibe una trama, se genera una interrupción en el sistema y ésta es atendida por la aplicación, la atención de las tramas que llegan implica que la aplicación está dedicada durante cierto tiempo a recibir la trama, a encapsularla y a realizar el procesamiento adecuado, todo esto lleva un determinado tiempo, que no es siempre el mismo para todas las tramas que aunque en principio el tráfico de datos que circula por la red de XBee va a ser bajo, pero puede darse el caso de que se iniciaran múltiples interrupciones en el sistema mínimamente espaciadas en tiempo, lo que puede suponer un problema, pues podría llegar a perderse información cuando el procesamiento individual es largo.

Para remediar este inconveniente la aplicación desarrollada implementa múltiples hilos de ejecución (*threads*), haciendo que la aplicación ejecute varios procesos al mismo tiempo (Pseudo-Paralelismo). Esto se consigue haciendo uso de las utilidades para concurrencia que Java ofrece, por lo que se hace uso de un “pool” de threads, el cual contiene instancias ya creadas y listas para usarse, cuando la aplicación recibe una trama recupera un thread(hilo de ejecución) de la pool, le asigna una tarea y lanza su ejecución, cuando la tarea termina el thread es liberado y devuelto a la pool. Con esto se consiguió reducir en lo posible el tiempo de respuesta de la aplicación, dando la capacidad de atender varias peticiones al mismo tiempo.

6.6 Escáner para la red ZigBee

Debido a que la aplicación necesita realizar la comunicación con el radio XBee, se desarrolló e implementó un conjunto de clases que realizan este proceso.

Para poder atender a los eventos generados por la recepción de tramas en el dispositivo XBee, la aplicación crea una instancia de la clase XBee la cual es capaz de gestionarlo, dando acceso a las interrupciones generadas por el puerto serie a la hora de recibir datos. La forma de instanciar la clase es la siguiente:

```
XBee xbee = new XBee();
```

Esta clase nos ofrece un mecanismo para manejar las interrupciones generadas por medio del método `addPacketListener()`. Teniendo lo anterior en cuenta se creó una clase que proporciona la funcionalidad de establecer un criterio común para el procesamiento de tramas, en forma de método abstracto, que implementan diferentes lectores concretos para las tramas que se reciben.

Lo siguiente a realizar consistió en leer y trabajar con las tramas recibidas, es aquí donde surge la problemática de una implementación genérica la cual debe procesar todas las tramas que lleguen sin importar de que tipo sean, además de que el procesamiento de dichas tramas debe ser común para todos los casos, esto en programación orientada a objetos se puede concretar con el uso de interfaces. Una interfaz es un contrato para todos aquellos que se adhieran a ella, es decir, deben implementar de forma concreta el comportamiento que la interfaz propone. Por lo que se ha creado una clase que proporciona la funcionalidad requerida para poderse ejecutar dentro de un *thread* al implementar *Runnable* y que además establece un criterio común para el procesamiento de tramas, en forma de método abstracto, que implementarán los lectores concretos para las tramas que se reciben.

```

/**
 * Clase base para los diferentes lectores especializados en tramas concretas.
 * Hace de plantilla para los que extiendan esta clase, que deben
 * implementar el método read(), para leer la trama recibida.
 * Implementa la interfaz Runnable, necesaria para el pool.
 */

public abstract class RFrameReader implements Runnable {

    private final XBeeResponse response;

    public RFrameReader(XBeeResponse response) {
        this.response = response;
    }

    public XBeeResponse getResponse() {
        return response;
    }

    protected abstract void read();

    public void run() {
        this.read();
    }

}

```

Esta clase proporciona una plantilla a través del método read(), para las clases que extienden de ella, es decir, read() es un método que implementará cada una de las clases que atiende un tipo especial de trama, estas instancias son generadas por la clase RFrameReaderFactory la cual gracias a la ayuda de la librería XBee-API que en el proceso de estructuración de la información recibida por parte del dispositivo crea un tipo concreto de trama como lo es RxResponse.

```

public class RFrameReaderFactory {

    private static final Logger log = Logger.getLogger(RFrameReaderFactory.class);

    public RFrameReader getResponseReader(XBeeResponse response) {
        if (response instanceof RxResponse || response instanceof ZNetRxResponse) {
            log.debug("Iniciando el lector StandardResponseReader");
            return new StandardFrameReader(response);
        } else if (response instanceof ModemStatusResponse) {
            log.debug("Iniciando el lector ModemStatusResponseReader");
            return new ModemStatusResponseReader((ModemStatusResponse) response);
        } else if (response instanceof ErrorResponse) {
            log.debug("Iniciando el lector ErrorResponseReader");
            return new ErrorResponseReader((ErrorResponse) response);
        } else if (response instanceof XBeeFrameIdResponse) {
            return new XBeeFrameIdResponseReader((XBeeFrameIdResponse) response);
        } else {
            log.warn("Respuesta desconocida - " + response.getApiId());
            throw new RuntimeException("Respuesta desconocida");
        }
    }

}

```

Con base en lo anterior se dispone de una forma de crear lectores especializados para las distintas especificaciones que DIGI ofrece en sus XBee, pero para este proyecto interesa la implementación de tramas del protocolo 802.15.4 tenemos el siguiente código.

```
public class StandardFrameReader extends RFrameReader {

    private static final Logger log = Logger.getLogger(StandardFrameReader.class);

    public StandardFrameReader(XBeeResponse response) {
        super(response);
    }

    public void read() {
        DataFormat dataFormat = new DataFormat(super.getResponse());
        IResponse response = dataFormat.parse();//proceso del mensaje
        if (response instanceof AssociationResponse) {
            log.debug("Recibida una trama de asociación");
            AssociationResponse associationResponse = (AssociationResponse)
dataFormat.parse();
            log.debug("Información de la asociación: " + associationResponse.toString());
        } else if (response instanceof ScanResponse64) { //se captura la trama scaneo de 64 bits
            log.debug("Recibida una trama de scan64");
            ScanResponse64 scanResponse64 = (ScanResponse64) response;
            log.debug("Informacion de la trama: " + scanResponse64);
            if (scanResponse64.getSignal() != "" && scanResponse64.getMacAddress() != null) {
                ScanPoint.add(scanResponse64);
            }
        } else if (response instanceof ScanResponse16) { //se captura la trama de scaneo de 16 bits
            log.debug("Recibida una trama de scanResponse16");
            ScanResponse16 scanResponse16 = (ScanResponse16) response;
            log.debug("Informacion de la trama: " + scanResponse16);
            if (scanResponse16.getSignal() != "" && scanResponse16.getNetAddress() != null) {
                ScanPoint.add(scanResponse16);
            }
        } else if (response instanceof MonitorResponse) { //se captura la trama de monitoreo
            log.info("Recibida trama de monitoreo");
            MonitorResponse monitorResponse = (MonitorResponse) response;
            log.info("Informacion de la trama: " + monitorResponse);
            if (monitorResponse.getSignal() != "" && monitorResponse.getAddress() != null) {
                ScanPoint.addMuestra(monitorResponse);
            }
        } else if (response == null) {
            log.info("No se puedo identificar la trama recibida");
        }
    }
}
```

Como se puede observar, la clase hereda efectivamente de RFrameReader, e implementa el método read() que lo que hace es evaluar el contenido de la trama para determinar si se trata de una trama de levantamiento o de una trama de monitoreo, para en cada caso realizar las acciones necesarias.

Para el caso de una trama de levantamiento (ScanResponse16), espera un contenido específico que identifique al dispositivo remoto al cual se le está realizando el levantamiento así como la potencia de recepción en los diferentes dispositivos baliza. Para el caso de una trama de monitoreo (MonitorResponse),

espera un contenido con las mediciones asociadas a los diferentes dispositivos baliza instalados en el edificio.

El contenido o *payload* de las tramas recibidas en la computadora tiene una estructura específica que nos permite definir los elementos característicos de las tramas de levantamiento y monitoreo.

6.7 Descripción de la aplicación

La aplicación desarrollada en este proyecto sirve como interfaz tanto para realizar el proceso de *fingerprinting* como para la localización del dispositivo. Dicha aplicación fue desarrollada en su totalidad sobre Java SE utilizando la librería java Swing.

6.7.1 Nuevo proyecto

Al abrir la aplicación se despliega un cuadro de dialogo, el cual nos pregunta si se desea abrir un archivo de datos ya existente o si se desea crear un nuevo proyecto. Si se elige “Crear un nuevo proyecto” se solicita seleccionar el archivo del mapa de la instalación a realizar el levantamiento.



Figura 6.3 Inicio de la aplicación.

6.7.2 Registro de dispositivos que conforman la red

Para la captura de datos de los dispositivos es mediante la opción dispositivos contenida dentro del menú vista, en la cual se puede agregar un nuevo dispositivo, modificar uno ya existente o eliminar alguno previamente capturado. Cabe destacar que éste debe ser el primer paso a realizar al ser un nuevo proyecto, ya que los datos de direcciones de red, posiciones de los dispositivos dentro de la construcción y los nombres de estos son estrictamente necesarios para poder realizar la comunicación entre la aplicación y los dispositivos que conforman el sistema.

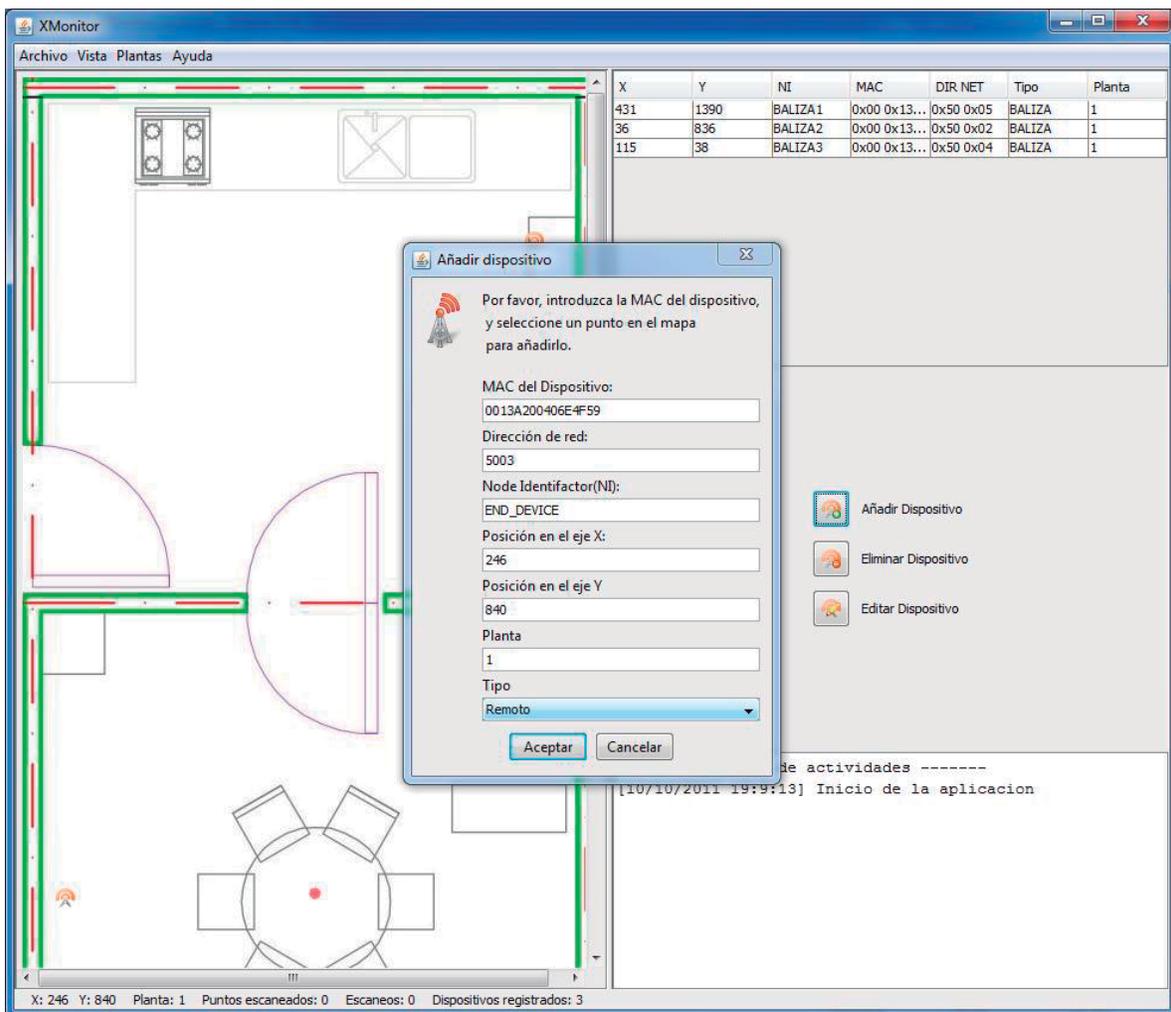


Figura 6.4 Captura de parámetros de configuración de los dispositivos.

6.7.3 Levantamiento (*fingerprinting*)

Una vez registrados los datos de los dispositivos en la aplicación, el programa permite, de una manera sencilla, realizar el proceso de *fingerprinting*, para esto se encienden todos los dispositivos y se despliegan los nodos baliza por la vivienda, a continuación, se sitúa el dispositivo remoto en la posición dentro de la vivienda en la que se desee conocer el vector de calibración, ahora desde la aplicación se selecciona la opción escanear contenida en el menú de vista y con el puntero seleccionamos en el mapa la posición que se va a escanear de tal manera que coincida gráficamente con la posición del dispositivo remoto dentro de la vivienda, entonces al pulsar sobre el botón de inicio de levantamiento representado por la antena de color azul, se realizarán tantos escaneos consecutivos como se haya fijado en el campo de escaneos. Los escaneos se realizan cada segundo. Para tomar una nueva medición simplemente se selecciona otro punto del mapa y se sitúa al dispositivo remoto en la nueva posición.

De esta manera, obtendremos una base de datos formada por un conjunto de tuplas del tipo: [*localización*, *vector de potencia*], donde *localización* hace referencia a las coordenadas (vector de dos dimensiones) de las posiciones de interés en localizar; y *vector de potencia*, al conjunto de valores RSSI de los nodos baliza cuando el nodo desconocido se encuentra en la localización correspondiente (vector cuya dimensión es el número de nodos baliza).

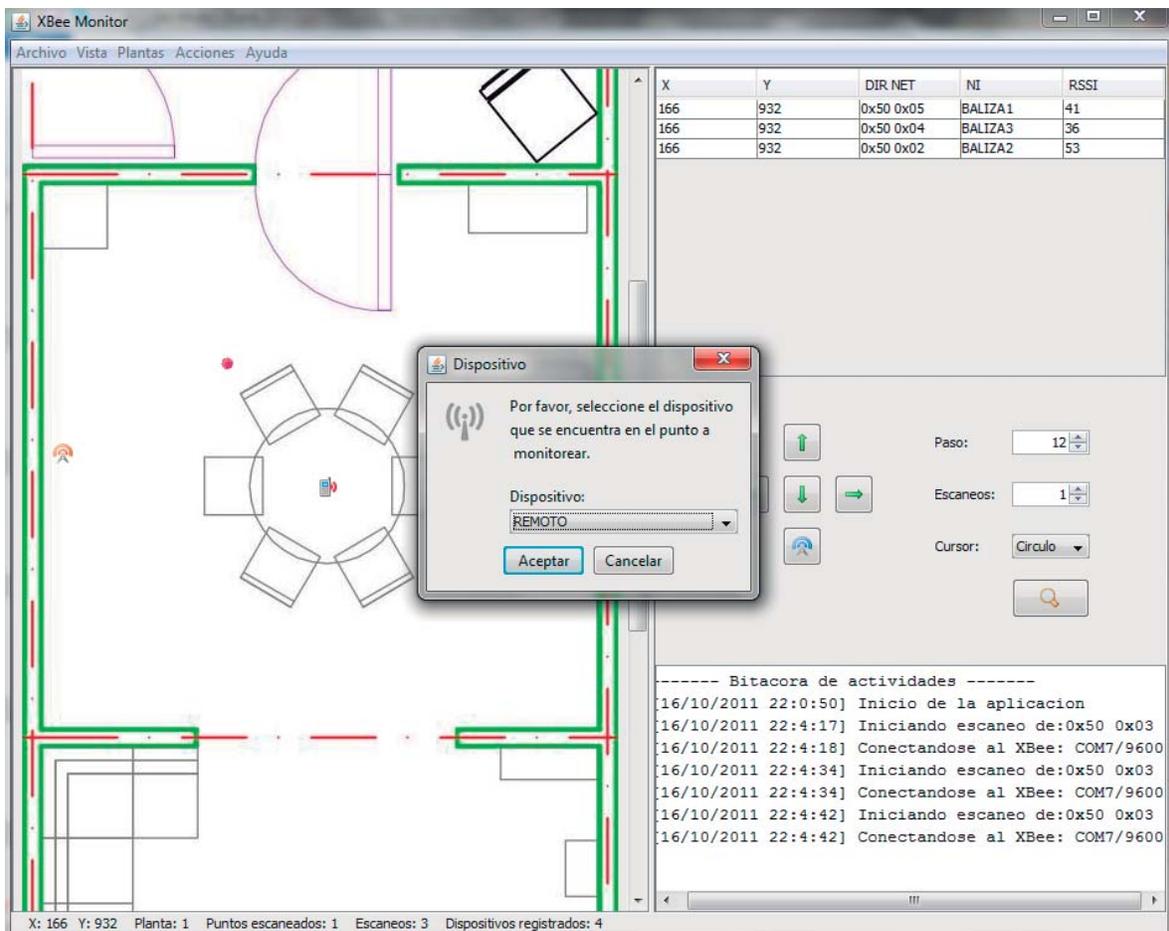


Figura 6.5 Realización del proceso de Fingerprinting.

Los datos obtenidos se muestran en la tabla superior de la aplicación, Figura 6.5, indicando la posición de escaneo, dirección de red, identificador de nodo (NI) y el parámetro RSSI. Además de que en el mapa de la izquierda, irán apareciendo en azul los puntos donde se ha escaneado.

La información del levantamiento es almacenada en un archivo en el directorio de ejecución del programa, este archivo guarda el estado del programa incluyendo datos del levantamiento e imágenes de los mapas. Si se desea guardar el archivo en un directorio específico, se puede utilizar la opción “guardar datos como...” del menú archivo.

6.7.4 Localización

Una vez que se han capturado los datos de los dispositivos y realizado el proceso de levantamiento, el siguiente paso es pasar a la vista “localización”, en donde se puede comenzar con el proceso de estimación de la posición, la aplicación da la opción de realizar la localización automática, la cual se repetirá cada cierto tiempo, parámetro definido por el usuario, o bien se puede forzar a realizar la localización en el momento que el usuario quiera, oprimiendo el botón de “localizar ahora”.

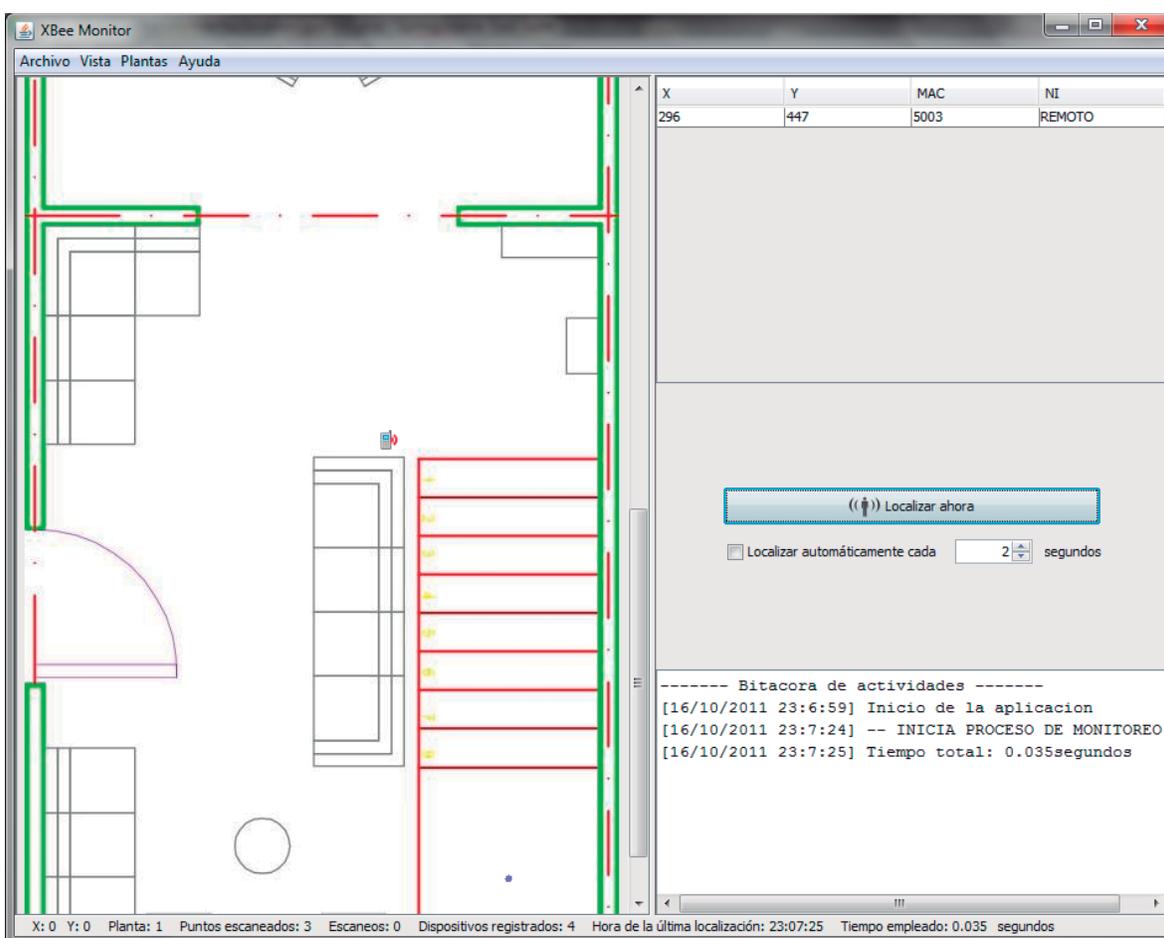


Figura 6.6 Localización del dispositivo remoto.

Al oprimir el botón de “localizar ahora” lo que se ejecuta es el escaneo de la red ZigBee, este proceso consiste en el envío de un paquete de datos a cada uno de los nodos baliza desplegados en la vivienda, en el cual se les indica que estos a su vez deben enviar un mensaje a un determinado dispositivo remoto, al recibir el

mensaje el dispositivo remoto toma la lectura del nivel de RSSI con el que se recibió dicho paquete y genera un nuevo mensaje de respuesta, en donde incluye el valor leído de RSSI, ahora el nodo baliza recibe el mensaje de respuesta del dispositivo remoto y nuevamente se toma la lectura del RSSI del paquete recibido, generando un nuevo mensaje en donde incluye el valor de la primera y segunda lectura del RSSI para finalmente enviar este paquete al radio que se encuentra conectado a la PC. A medida que se van recibiendo los mensajes de los nodos baliza, se reenvían por medio de la conexión USB a la computadora para que procese la información y estime finalmente la ubicación del nodo desconocido.

Como cada mensaje que es recibido contiene dos lecturas de RSSI, éstas son promediadas para tener un solo dato. El motivo por el que se toma el valor del RSSI en dos partes distintas del proceso de localización es la extremada variación de las lecturas. Considerar el valor RSSI de dos mensajes reduce el efecto que produce la variación de las medidas de potencia de un mensaje a otro, lo que es tomado en cuenta es la media de las lecturas y no cada uno de forma individual.

Los datos obtenidos durante el proceso de localización son procesados por el algoritmo antes mencionado, éste debe comparar los vectores de potencia recibidos durante el funcionamiento de la aplicación con los almacenados en la base de datos durante la fase de calibración (levantamiento). De esta manera, realizando un simple cómputo en el que se escoja el vector de potencia de la base de datos de menor distancia euclidiana al vector de potencia obtenido, se puede estimar la posición del nodo desconocido.

Los motivos por los que se ha realizado este diseño están relacionados con las limitaciones computacionales de los nodos XBee. Por lo que, se ha optado por llevar a cabo la mayor carga computacional en la estación base (PC).

7. Pruebas y resultados

El objetivo de este apartado es describir el proceso que se llevó a cabo para estimar la localización del nodo móvil (nodo desconocido) a partir de una serie de nodos baliza (nodos fijos).

Para la realización de las pruebas se cuentan con 5 nodos, 3 de los cuales actuarán como balizas, uno como nodo desconocido y finalmente el último actuará como la interfaz entre la PC y los dispositivos XBee. El sistema se probó en una edificación de dimensiones 4.5 [m] x 16 [m] (72 [m²]), el plano de dicho edificio se observa en la Figura 7.1; La prueba consistió en instalar una red ZigBee, usar el programa de localización desarrollado y probar el funcionamiento del sistema al localizar un dispositivo remoto.

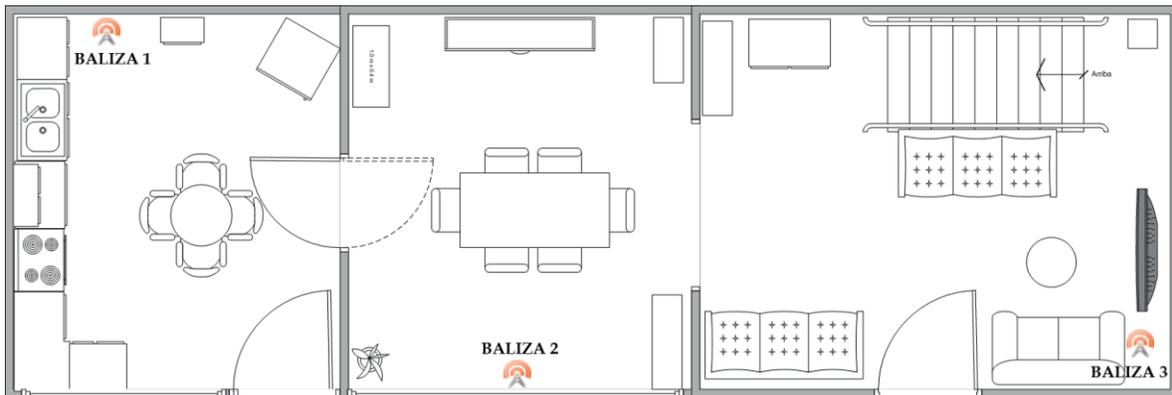


Figura 7.1 Plano de la Planta.

Los nodos baliza se han distribuido como se observa en la Figura 7.1, así en toda la construcción existirá señal proveniente de los tres nodos baliza. Todos los radios se colocaron a una altura de 1.1 [m] sobre el nivel del suelo para tratar de evitar la mayoría de los obstáculos presentes en un entorno habitual.

Una vez configurados todos los elementos que intervienen en el sistema de localización, se procede a realizar el levantamiento, para lo cual se ejecuta el programa desarrollado y se crea un nuevo proyecto siguiendo los pasos descritos en la sección 6.7.

La fase de entrenamiento se realizó utilizando el módulo XBee-Pro con NI: *remoto*, y se obtienen las posibles posiciones de dicho módulo (marcas azules) dentro del plano, las cuales son mostradas en la Figura 7.2, éstas se han tomado con un espaciamiento mínimo de 3 metros o en su defecto en posiciones sin línea de vista tratando de que exista una variación notable del RSSI entre los distintos puntos.

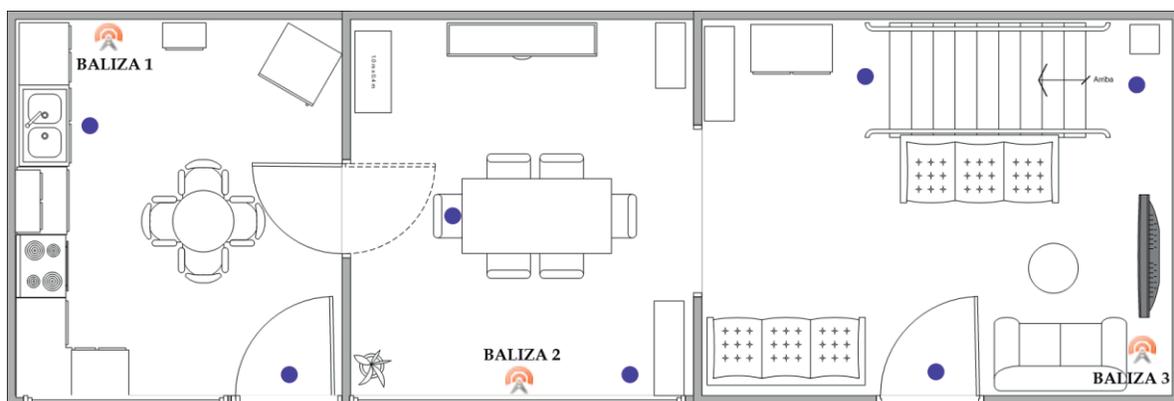


Figura 7.2 Posiciones del dispositivo remoto.

Cabe señalar que en cada punto del levantamiento se tomaron medidas del RSSI girando el dispositivo en intervalos de 30° , por lo cual para un mismo punto se tomaron 12 posibles posiciones y en cada uno de estas se tomaron 5 lecturas del valor RSSI, medida que se tomó debido a que al realizar las pruebas se pudo

observar que a determinados ángulos se tenía variaciones grandes en la lectura de la potencia recibida, detectándose variaciones de hasta 3 [dBm].

Después del entrenamiento se ejecuta el programa de localización contenido en el menú vista, el cual utiliza la información generada durante la fase de entrenamiento permitiendo el desplazamiento del módulo móvil dentro de la construcción y determinando la posición de referencia más cercana. La precisión de la localización varía con el número de puntos de acceso, la posición de éstos y el número de puntos de referencia. De la manera en que se ha establecido la prueba, se puede obtener una exactitud de aproximadamente 3[m] de radio, distancia que dependiendo de la aplicación puede ser aceptable o no.

Dadas las características particulares de esta aplicación, los requerimientos de exactitud en las localizaciones no son muy estrictos en cuestión de requerir localizaciones en términos de centímetros. Más que disponer de una información muy precisa sobre la posición del objeto, se busca conocer en qué habitación se encuentra en cada momento. De este modo, puede en trabajos futuros enviarse una alarma avisando que el dispositivo lleva más de un determinado tiempo en una habitación de la vivienda o que simplemente este se encuentra en un área prohibida. Así, podemos afirmar que la aplicación implementada no acarreará un costo computacional alto ya que no es necesaria una exactitud de centímetros en estricta palabra.

7.1 Colocación de los nodos baliza.

Con el fin de dotar de cobertura a toda la vivienda, se situó un nodo baliza en cada una de las habitaciones. La ubicación de los nodos es un aspecto clave, ya que influirá en la precisión de los resultados, se realizaron diferentes pruebas las cuales se citaran a continuación.

Inicialmente se pensó en colocar los nodos de modo que los nodos baliza permanecieran en algunos muebles propios del entorno de tal manera de que no se necesitara realizar modificaciones al mobiliario, dado lo anterior los nodos quedaron colocados a diferentes alturas con respecto al piso, solo se cuidó que el lugar en donde quedaban instalados fuera lo suficientemente seguro para evitar accidentes eléctricos o pudiera causar daño a las personas que transitaran por la habitación.

Al poner en práctica el sistema se conseguían localizaciones poco exactas e imprecisas. Esto puede deberse fundamentalmente a que el plano en el que se encontraba cada nodo baliza era distinto de donde se situaba el dispositivo remoto, provocando que los planos de máxima radiación de los nodos no coincidiesen. Esta situación provocó que en la mayoría de los casos la señal recibida en cada uno de los nodos fuera debida principalmente a componentes que han sufrido el efecto de multitrayecto, haciendo que la variabilidad en las medidas aumente.

Al repetir las pruebas situando todos los nodos baliza en el centro de cada habitación y en el plano del nodo desconocido (a mediana altura para evitar la mayoría de los obstáculos), la exactitud y la precisión en la ubicación del nodo desconocido se incrementaron de forma notable.

No obstante, dado que no es práctico colocar los nodos baliza en el centro de las habitaciones, se repitió de nuevo el experimento colocando los nodos baliza en las esquinas de las habitaciones, con la diferencia de que en esta ocasión los situamos en el mismo plano que el nodo desconocido. La exactitud obtenida con este despliegue era muy similar al caso en el que los nodos baliza estaban situados en el centro de las habitaciones a mediana altura. De este modo se pudo observar que el plano en el que se sitúan los nodos es un factor importante en la exactitud del sistema, siendo este más trascendente que la propia ubicación de los nodos dentro de la habitación. Así, la ubicación definitiva de los nodos baliza es la

mostrada en la Figura 7.1, tomando en cuenta que todos los nodos se encuentran a la misma altura con respecto del piso.

7.2 Mapa de cobertura

La prueba se ha realizado situando el dispositivo remoto en distintos lugares de cada habitación, tomando las lecturas del RSSI a una altura de 1.1 [m] del suelo. La Figura 7.3 muestra los puntos donde se ha realizado cada escaneo (marcas azules). Estas medidas son las que hay que analizar para realizar la estimación de la localización, tal como se explicó en el Apartado. En la Tabla 7-1 se indican los valores numéricos.

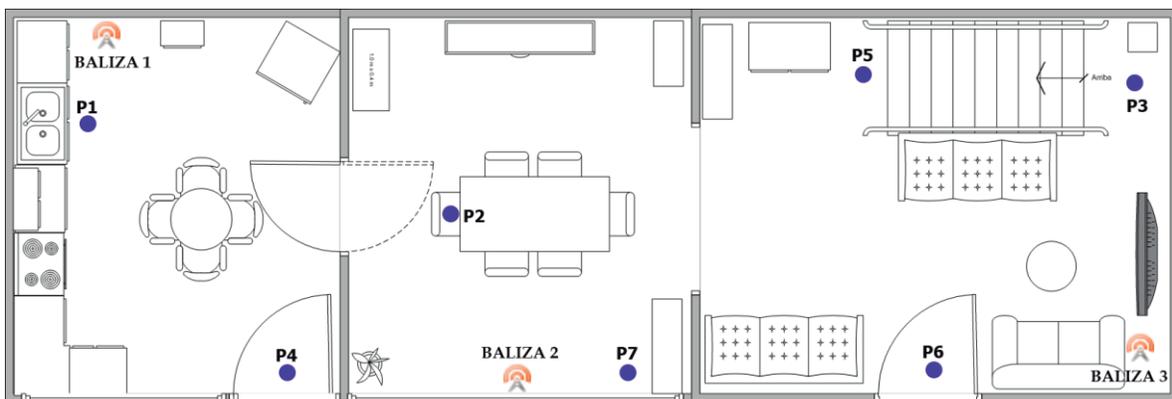


Figura 7.3 Puntos de escaneo.

Tabla 7-1 Vectores de Potencia.

Habitación 1(cocina)		
P ₁ =[38,52,45]	P ₄ =[39,52,39]	
Habitación 2(comedor)		
P ₂ =[39,47,36]	P ₇ =[46,44,36]	
Habitación 3(Sala)		
P ₃ =[53,42,50]	P ₅ =[57,37,47]	P ₆ =[56,50,41]

7.3 Consideraciones / limitaciones

Si bien el algoritmo KNN, puede parecer atractivo debido al bajo costo computacional requerido, este presenta algunas limitaciones, ya que pueden ocurrir situaciones como la mostrada en la Figura 7.4, en la que, pese a que el nodo desconocido está ubicado claramente en la posición B, la aplicación estimará que se encuentra en la posición A, esto es debido a que la atenuación introducida por el espacio entre los dos puntos no es lo suficientemente grande como para obtener una diferencia notable en el valor RSSI que pudiera diferenciar ambas posiciones.

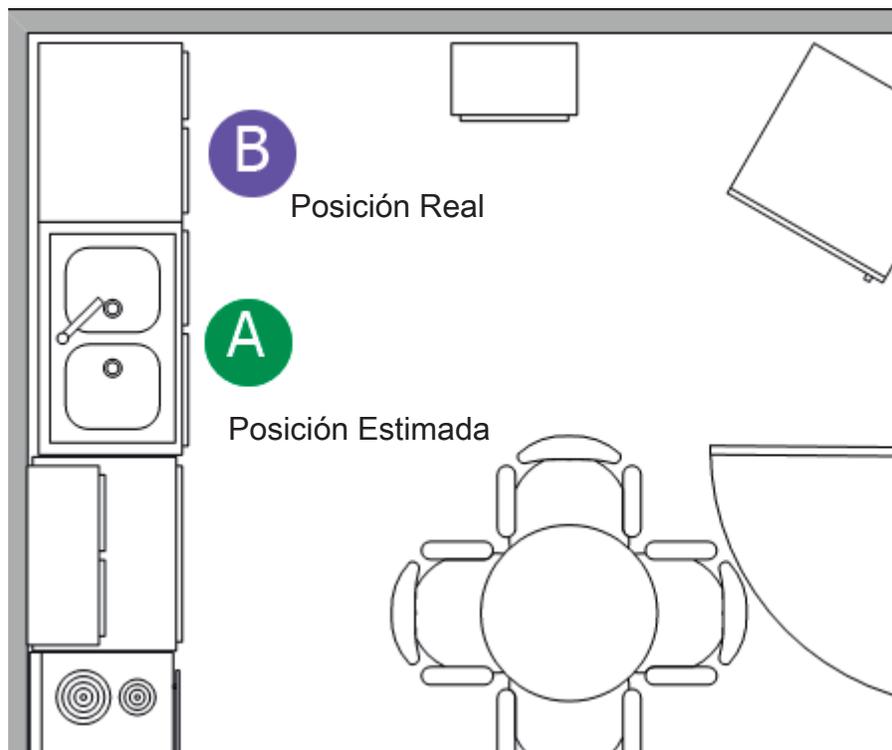


Figura 7.4 Limitaciones de la aplicación.

8. Conclusiones

Los sistemas de posicionamiento en interiores actualmente se encuentran en estado de desarrollo, se ha expuesto en capítulos previos, se han realizado grandes avances utilizando distintas tecnologías, aunque aún no hay una tecnología que resuelva por completo el problema.

En el desarrollo del proyecto se trató de conseguir una tecnología que pueda ser comercializada fácilmente y al mismo tiempo sea funcional, cumpliendo con los requisitos funcionales del sistema planteado. Con la tecnología elegida se ha conseguido el objetivo previsto y al mismo tiempo se ha obtenido una plataforma que se puede implantar fácilmente en cualquier edificación.

La herramienta de localización desarrollada, tal y como se ha detallado en el trabajo, utiliza el modelo de localización *Fingerprinting*, el cual proporciona una precisión bastante aceptable teniendo en cuenta los entornos para los cuales está destinada la aplicación.

El modelo de localización aplicado a los requerimientos de precisión de este trabajo es el más apropiado si tenemos en cuenta la relación costo de implementación/costo de cálculos/precisión. No obstante, se encontraron algunos inconvenientes dignos de reseñar:

- La existencia de fluctuaciones de la señal recibida en un lugar determinado a lo largo del tiempo introducen una serie de saltos indeterminados en el proceso de localización. Estos saltos pueden producir que la localización varíe alrededor de un punto.

- Puede darse el caso de que el dispositivo remoto obtenga en varios puntos del edificio valores similares de RSSI de “x” nodos balizas accesibles todos ellos desde dichos puntos, lo que provocaría que el algoritmo pudiera situarnos en cualquiera de esos puntos.
- Otro inconveniente es que con este método no se puede trazar correctamente la trayectoria real que sigue un usuario durante el movimiento a través del edificio. Por lo que se podrían obtener localizaciones físicamente imposibles, como causa de la velocidad de desplazamiento del dispositivo cliente.
- Los sistemas de localización implementados con la tecnología ZigBee no sirven para estimar localizaciones de precisión de centímetros.
- Además, los nodos tienen capacidades computacionales limitadas, por lo cual no son capaces de ejecutar algoritmos de localización complejos.
- La mayor limitación del método *fingerprinting* es que requiere que el espacio radioeléctrico sea lo más estable posible, si la naturaleza del entorno es dinámico no es posible aplicarlo.

A pesar de estos inconvenientes mencionados, las ventajas que se encuentran son las siguientes:

- El algoritmo de localización es lo suficientemente eficiente como para que la capacidad de cómputo requerida sea asumible por computadoras comerciales con características estándar, evitando la necesidad de equipos muy sofisticados.
- Otro beneficio es que, para aumentar considerablemente el rendimiento del algoritmo de localización, podemos incrementar la densidad de nodos baliza del sistema, lo cual, debido al estado actual del mercado, no implicaría un gasto sencillo, pero si la aplicación requiere de una mayor exactitud en la localización de los dispositivos que conforman el sistema, el gasto es factible. Sin embargo, una densidad excesiva de puntos de acceso aumentaría ligeramente el tiempo de respuesta del sistema.

- Con el software estándar de cualquier sistema operativo y las librerías incluidas en la aplicación desarrollada es suficiente para su correcto funcionamiento, por lo que no es necesario ningún tipo especial de requerimientos de software.
- Otra gran ventaja del proyecto fue el uso del estándar 802.15.4, el cual facilitó la creación de una red WPAN al contar ya con el sistema de anticolidión CSMA/CA y de envío de acuse de recibido de trama (ACK), asegurando que la información llegara correctamente al receptor, esto brinda cierta flexibilidad al crear diversas aplicaciones sin la necesidad de profundizar en el conocimiento de la capa física. Además, el reducido ancho de banda que tiene esta tecnología no limita la cantidad de aplicaciones que se pueden crear, los sistemas de monitoreo y la domótica en general, son los mejores ejemplos de su gran utilidad.
- El software del proyecto ha sido desarrollado pensando en futuras extensiones del sistema, de modo que pueda ser aplicado a diferentes ámbitos donde la localización podría ser útil, agregando nuevas funcionalidades como el monitoreo de variables ambientales, control de acceso y restricción de zonas, solo por mencionar algunas.
- El empleo de Java SE como plataforma de desarrollo fue un éxito, la combinación de una API accesible y bien documentada junto con la implementación y despliegue de la tecnología XBee permite prever que en un futuro cercano, tanto el número de aplicaciones como la necesidad de desarrolladores en este ámbito aumente, por lo que la experiencia adquirida durante el desarrollo de este proyecto será muy valorada.

Es evidente que los niveles de RSSI son la forma más económica y rápida de proveer un mecanismo de localización a los radios XBee, ya que el nivel de RSSI no es un parámetro adicional que haya que añadir, si no que se obtiene de los propios mensajes que se intercambian. Sin embargo, hay que tener presente que el entorno radioeléctrico es cambiante, y ni aún centrándonos en un mismo punto se obtiene una lectura constante. Por este motivo se debe realizar un estudio

previo de cómo dividir en sectores, de tal manera que el algoritmo tuviese la menor incertidumbre posible a la hora de estimar la posición, consiguiendo una mayor precisión

Se ha comprobado que para llevar a cabo una correcta localización del nodo desconocido, es necesaria una adecuada colocación de los nodos baliza (todos a la misma altura y libres de obstáculos que los cubran) y una fase de calibración que represente al entorno bajo estudio de la forma más fiable posible. La fase de calibración es vital para que la localización sea efectiva, ya que si la base de datos obtenida en este proceso no es lo suficientemente representativa del entorno, la localización del nodo sería equívoca.

La técnica desarrollada arroja buenos resultados siempre que las condiciones de trabajo en la fase de calibración y durante el funcionamiento de la aplicación de localización sean lo suficientemente similares, es decir, en el momento en el que se produzcan cambios significativos en el entorno (cambio de posición de mobiliario de grandes dimensiones, construcción de nuevas paredes, etc.) habría que actualizar la base de datos.

Con el proyecto descrito a lo largo de este documento no solo se logró la obtención de aprendizajes nuevos, también se tuvo la oportunidad de utilizar los conocimientos adquiridos a lo largo de mi formación profesional, sobre todo los relacionados con las tecnologías inalámbricas, programación en Java y C, manejo de multitareas, e incluso con temas no conocidos en ese momento, se tuvo que hacer acopio de todos los recursos posibles incluyendo, procesos de ensayo-error, hasta conseguir con el resultado deseado.

Este proyecto tiene una amplia aplicación comercial, ya que este es una buena base para crear un sistema de monitoreo con todas las condiciones que el mercado exige. Además de que la aplicación ha servido para demostrar que el posicionamiento en interiores mediante el protocolo IEEE 802.15.4 es posible, permitiendo mejorar en un futuro su precisión y exactitud mediante el empleo de

algoritmos más complejos y otras técnicas como el preprocesado o postprocesado de datos.

8.1 Líneas futuras de trabajo.

En el transcurso de este trabajo se encontraron diversas dificultades a la hora de alcanzar los objetivos planteados. Sin embargo, durante el tiempo que transcurrió el desarrollo de éste, se fueron detectando algunas mejoras y complementos.

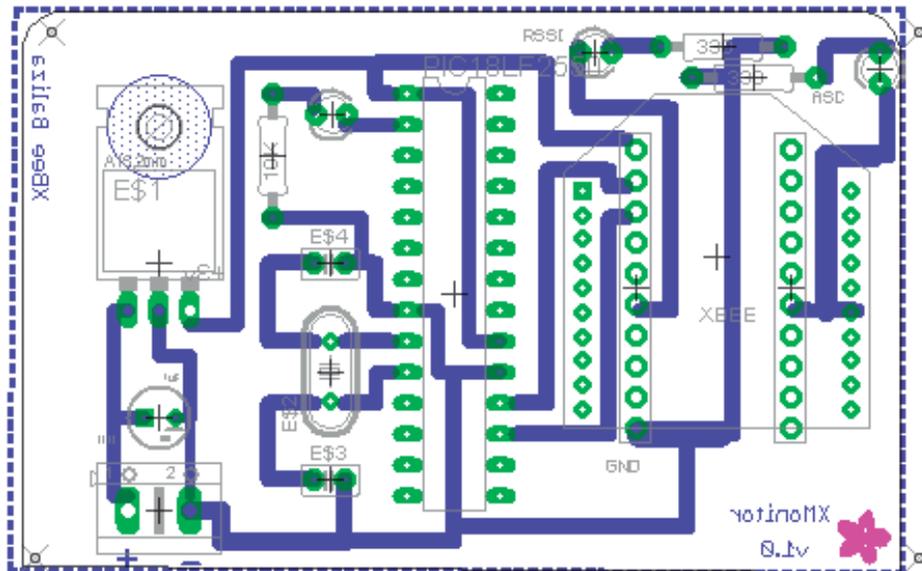
Si bien en este proyecto se ha centrado principalmente en el proceso de localización, es posible añadir diversas funcionalidades a la aplicación de un modo sencillo. Podríamos destacar las correspondientes al monitoreo de variables físicas del entorno como son: el control de la temperatura y humedad, lo que posibilita la detección de un foco de incendio en cualquier punto de una vivienda, o simplemente el monitoreo de estos parámetros; medidas de luminiscencia y el uso de acelerómetros, podrían emplearse para situar los nodos en un estado de reposo, de modo que se reduzca el gasto de energía cuando el sistema lo determine esto conveniente en base a estos dos parámetros censados y no sea necesaria la estimación de su posición.

Entre otros puntos a considerar para perfeccionar la aplicación de localización son los siguientes:

- Información de la vivienda: si se tuviese en cuenta que no es posible cualquier combinación en cuanto a cambios de localización se refiere, podríamos mejorar la aplicación para que eliminase posibilidades de localización que no se pueden dar en la práctica. Por ejemplo, en nuestro caso no es posible estar en la cocina sin pasar antes por el comedor.
- Mayor número de referencias: con un mayor número de nodos baliza, la estimación de la posición sería más precisa ya que tendríamos mayor número de referencias para su cómputo en el entorno bajo estudio

- Enviar los avisos y alarmas: realizar notificaciones por medio del sistema de telefonía móvil así como por medio de correo electrónico. De tal manera que se tenga una alarma para el caso de no poder obtener la localización de un dispositivo remoto.

Placa fenolica



Código fuente

```
void main(void){
    int ret;
    int tmp;
    static APIstruct apirx;
    static APIdata rxdata;
    APIinfo *info;
    APIrecv16 *r16;
    APIresponse *res;
    XBeePacket myPacket, *packet;
    XBeePacket myPacket1, myPacket2;
    packet = &myPacket;
    setup_uart(9600);
    enable_interrupts(INT_RDA);
    enable_interrupts(GLOBAL);
    API_initgettingframe(&apirx); //comienza recepción
    buf_index = 0;
    while(TRUE){
        output_toggle(LedR);
        buf_index = 0;
        while((ret=API_getframe(&apirx,escape))<=0); //recibe trama
        rxdata.len=ret; //obtiene longitud de datos
        memcpy(&rxdata.info,&apirx.frame.info,ret); //copia, libera buffer
        API_initgettingframe(&apirx); //recibe trama siguiente (luego)
```

```

info=&rxdata.info;

XBee_ResetPacket( &myPacket1 );
while(XBee_GetPacket(&myPacket1) == 0){
    delay_ms(100);
}
packet= &myPacket1;
info->id = packet->txStatus.frameID;
switch(info->id){
    //Tipo de trama
    case API_APIRECV16: //DATA, dirección de 16-bits
        r16=(APIrecv16 *)info->data; //apunta a DATA
        if( info->data[4] == 0x01 ){
            //envió hacia el end device
            packet->apiId = XBEE_COMM_TX16;
            packet->tx16.frameID = 0x00;
            packet->tx16.destination[0] = info->data[5];
            packet->tx16.destination[1] = info->data[6];
            packet->tx16.options = 0x00;
            packet->tx16.data[0] = 0x02;
            XBee_SendPacket( &mypacket, 1 );
        }else if( info->data[4] == 0x02 ){
            //recepción en el end device y repuesta a la
            //radio base
            packet->apiId = XBEE_COMM_TX16;
            packet->tx16.frameID = 0x00;
            packet->tx16.destination[0] = r16->addr[0];
            packet->tx16.destination[1] = r16->addr[1];
            packet->tx16.options = 0x00;
            packet->tx16.data[0] = 0x03;
            packet->tx16.data[1] = r16->rssi;
            XBee_SendPacket( &mypacket, 2 );
        }else if( info->data[4] == 0x03 ){
            //recepción en radio base y envió
            //de información al coordinador
            packet->apiId = XBEE_COMM_TX16;
            packet->tx16.frameID = 0x00;
            packet->tx16.destination[0] = 0x50;
            packet->tx16.destination[1] = 0x00;
            packet->tx16.options = 0x00;
            packet->tx16.data[0] = 0x04;
            packet->tx16.data[1] = info->data[5];
            packet->tx16.data[2] = r16->rssi;
            XBee_SendPacket( &mypacket, 3 );
        }else if( info->data[4] == 0x04 ){
            //Recepción en el end device y envió de paquete a
            //la baliza
            packet->apiId = XBEE_COMM_TX16;
            packet->tx16.frameID = 0x00;
            packet->tx16.destination[0] = info->data[5];
            packet->tx16.destination[1] = info->data[6];
            packet->tx16.options = 0x00;
            packet->tx16.data[0] = 0x05;
            XBee_SendPacket( &mypacket, 1 );
        }else if( info->data[4] == 0x05 ){
            //Recepción en la baliza y envió de la información
            //hacia el coordinador
            packet->apiId = XBEE_COMM_TX16;

```

```
        packet->tx16.frameID = 0x00;
        packet->tx16.destination[0] = 0x50;
        packet->tx16.destination[1] = 0x00;
        packet->tx16.options = 0x00;
        packet->tx16.data[0] = 0x06;
        packet->tx16.data[1] = r16->rssi;
        XBee_SendPacket( &mypacket, 2 );
    }
    break;
default:
    break;
}
input_ready = 0;
}
```

9. Bibliografía

- Información acerca del lenguaje Java: <http://java.sun.com>
- **Cánovas López, Andrés**, *Manual de usuario del compilador PCW de CCS*® (C Compiler for Microchip PICMicro MCUs).
- *Manual de usuario X-CTU*, 2008, Digi International
- **Shashank Tadakamadla**, *Indoor Local Positioning System for ZigBee, Based on RSSI*, Thesis Report, MID Sweden University, 2006
- Java API: <http://java.sun.com>
- **Allan Hollub**, “*Programming Java Threads in the Real World*” (Parts 1-9) : <http://www.javaworld.com>
- **García Breijo, Eduardo**, *Compilador C CCS y Simulador Proteus Para Microcontroladores PIC*.
- **Deitel Deitel**, *Como Programar en Java*. 5ta Edición
- Redes de sensores: <http://zone.ni.com/devzone/cda/tut/p/id/9507>
- Localización en redes ZigBee:
<http://www.priorartdatabase.com/IPCOM/000141520/>
- **Gaona, A.L.** *Introducción al desarrollo de programas con Java*. s.l. : UNAM, 2007
- **MaxStream, Inc.**, *XBee™/XBee-PRO™ OEM RF Modules Product Manual v1.xAx - 802.15.4 Protocol*, www.MaxStream.net (live chat support)
- **Andrés Oyarce, Paul Aguayo**. *Guía del Usuario XBee Series 1*, Ingeniería MCI LTDA.
- **Microingenia**, S.L. Creative Commons BY NC SA. *ModXbee: Módulo comunicaciones Xbee de MaxStream*.
- **Telmo Moya; Daniel Hoyos**, *Gateway USB para el control de módulos Xbee con MODBUS RTU*. INENCO – Instituto UNSa